

---

# StreamUPC - iOS

## UN NOU CONCEPTE DE COL·LABORACIÓ

---

### TREBALL FINAL DE GRAU

AUTOR

NÉSTOR MALET MONTOLÍO

DIRECTOR:

JAUME MORAL ROS



CO-DIRECTOR:

ALBERT OBIOLS VIVES



PONENT:

JOSEP CASANOVAS  
GARCIA

DEPARTAMENT EIO

16 DE DESEMBRE DEL 2013

GRAU EN ENGINYERIA INFORMÀTICA  
ENGINYERIA DEL SOFTWARE

FACULTAT D'INFORMÀTICA DE BARCELONA  
UNIVERSITAT POLITÈCNICA DE CATALUNYA



*M'agradaria agrair al Jaume Moral i l'Albert Obiols per la seva ajuda i supervisió, així com per haver-me ofert la possibilitat de realitzar aquest projecte.*

*També m'agradaria agrair a tot l'equip SomUPC, Albert, Jaume, David Martínez, Albert Almà i Eric Lopez. Per la vostra ajuda i suport. He gaudit molt treballant en aquest projecte i sense vosaltres no hauria estat possible. Especialment al David Martínez i l'Albert Almà per haver-me aguantat durant la redacció d'aquesta memòria.*

*A més de l'equip SomUPC, el projecte no hauria estat possible sense UPCnet (Javier Otero, Janet Durà, Marcos Montero, Víctor Fernández, Carles Bruguera, Marc Calvet i Oriol Bosch) ni sense l'inLab FIB.*

*La redacció d'aquesta memòria hagués estat impossible sense l'ajuda del Marc Comino, la Montse Jándula i la Sandra Crespo.*

*Per descomptat, també he d'agraciar el suport de la meua família i tots els meus amics, no només durant aquest treball, sinó durant tota la carrera.*

## **Abstract (Català)**

Els sistemes d'informació actuals de la Universitat Politècnica de Catalunya no tenen en compte l'aspecte comunitari. El client del projecte, UPCnet, està desenvolupant un conjunt de serveis per fomentar la col·laboració i millorar la comunicació entre els membres de la comunitat UPC, utilitzant les últimes innovacions tecnològiques.

Aquests nous serveis estan centrats en el MAX. El MAX és un servidor d'activitat social, en el que els usuaris publiquen i consulten contingut. En aquest treball final s'ha desenvolupat una aplicació iOS client del MAX.

L'aplicació permet a l'usuari consultar l'activitat social generada pels altres usuaris, generar nova informació i permet comunicar-se amb els altres usuaris o grups utilitzant converses privades.

Els serveis estan pensats per a la comunitat UPC, però el client està implantant els productes a altres comunitats que també ho han adquirit, per exemple empreses o col·lectius de professionals.

## Abstract (Español)

Los sistemas de información actuales de la *Universitat Politècnica de Catalunya* no tienen en cuenta el aspecto comunitario. El cliente del proyecto, UPCnet, está desarrollando un paquete de servicios para fomentar la colaboración y mejorar la comunicación entre los miembros de la comunidad UPC, utilizando las últimas innovaciones tecnológicas.

Estos nuevos servicios están centrados en el MAX. El MAX es un servidor de actividad social, en el que los usuarios publican y consultan contenido social. En este trabajo se ha desarrollado una aplicación iOS cliente del MAX.

La aplicación permite al usuario consultar la actividad social generada por los otros usuarios, generar nueva información y comunicarse con los otros usuarios o grupos mediante las conversaciones privadas.

El paquete de servicios está pensado para la comunidad UPC, pero el cliente está implantando los productos en otras comunidades que también lo han adquirido, por ejemplo empresas o colectivos de profesionales.

## Abstract (English)

The *Universitat Politècnica de Catalunya*'s information systems do not address community issues good enough. UPCnet, the project's client, is developing a suite of services to help promote collaboration and improve communication between the members of the UPC community.

These systems are centered in the MAX server which is an activity stream server, where users publish and obtain social content. The goal of this project has been the development of the MAX client iOS app.

Users can see the social activity generated by others, share new information and communicate with each other or with groups of users through private chats.

These services are designed for the UPC community, but the client is also deploying the product to communities that have acquired it, like companies or professional collectives.

# Índex

|   |            |
|---|------------|
| <b>Agraïments</b>                                     | <b>I</b>   |
| <b>Abstract (Català)</b>                              | <b>II</b>  |
| <b>Abstract (Español)</b>                             | <b>III</b> |
| <b>Abstract (English)</b>                             | <b>IV</b>  |
| <b>1 Introducció</b>                                  | <b>7</b>   |
| <b>2 Gestió del projecte</b>                          | <b>9</b>   |
| 2.1 Metodologia . . . . .                             | 9          |
| 2.1.1 Iteracions - <i>Sprints</i> . . . . .           | 10         |
| 2.1.2 <i>Daily Scrum</i> . . . . .                    | 10         |
| 2.1.3 Tauler de tasques - <i>Task board</i> . . . . . | 11         |
| 2.1.4 Història d'usuari . . . . .                     | 12         |
| 2.1.5 Control de versions . . . . .                   | 13         |
| 2.2 Planificació inicial . . . . .                    | 15         |
| 2.2.1 Recursos temporals . . . . .                    | 15         |
| 2.2.2 Fases del projecte . . . . .                    | 15         |
| 2.2.3 Seguiment de la planificació . . . . .          | 18         |
| 2.2.4 Diagrama de Gantt . . . . .                     | 18         |
| 2.3 Planificació final . . . . .                      | 21         |
| 2.4 Pressupost . . . . .                              | 22         |
| 2.4.1 Recursos humans . . . . .                       | 22         |
| 2.4.2 Recursos Materials . . . . .                    | 23         |
| 2.4.3 Costos indirectes . . . . .                     | 24         |
| 2.4.4 Cost total . . . . .                            | 24         |
| 2.4.5 Control del pressupost . . . . .                | 25         |
| 2.4.6 Viabilitat econòmica . . . . .                  | 25         |
| 2.5 Sostenibilitat i Responsabilitat Social . . . . . | 26         |
| 2.5.1 Control de la sostenibilitat . . . . .          | 26         |
| 2.6 Lleis i regulacions . . . . .                     | 27         |
| <b>3 Estudi prèvi</b>                                 | <b>29</b>  |
| 3.1 Context . . . . .                                 | 29         |
| 3.1.1 UPCnet . . . . .                                | 29         |
| 3.1.2 inLab FIB . . . . .                             | 29         |
| 3.1.3 Director i co-director . . . . .                | 30         |
| 3.1.4 Autor . . . . .                                 | 30         |
| 3.1.5 Usuari final . . . . .                          | 30         |
| 3.2 Objectius . . . . .                               | 31         |
| 3.3 Estat de l'art . . . . .                          | 32         |
| 3.3.1 Entorn de col·laboració . . . . .               | 32         |
| 3.3.2 Innovació . . . . .                             | 33         |
| 3.4 Abast . . . . .                                   | 35         |

|          |   |           |
|----------|---|-----------|
| 3.5      | Mètodes de validació . . . . .              | 36        |
| 3.6      | Riscs . . . . .                             | 37        |
| 3.7      | Definició de termes i acrònims . . . . .    | 38        |
| <b>4</b> | <b>Especificació</b>                        | <b>39</b> |
| 4.1      | Requisits funcionals . . . . .              | 39        |
| 4.1.1    | Històries d'usuari . . . . .                | 39        |
| 4.1.2    | Model conceptual . . . . .                  | 49        |
| 4.1.3    | Model de comportament . . . . .             | 50        |
| 4.2      | Requisits no funcionals . . . . .           | 51        |
| <b>5</b> | <b>Disseny arquitectònic</b>                | <b>55</b> |
| 5.1      | Patrons de disseny . . . . .                | 55        |
| 5.1.1    | Model Vista Controlador . . . . .           | 55        |
| 5.1.2    | Notificacions . . . . .                     | 56        |
| 5.1.3    | Categoria . . . . .                         | 57        |
| 5.1.4    | Delegat . . . . .                           | 57        |
| 5.1.5    | Singleton . . . . .                         | 58        |
| 5.2      | Protocols . . . . .                         | 59        |
| 5.2.1    | <i>Activity Streams</i> . . . . .           | 59        |
| 5.2.2    | REST . . . . .                              | 59        |
| 5.2.3    | STOMP . . . . .                             | 60        |
| 5.3      | Capa de presentació . . . . .               | 61        |
| 5.3.1    | Disseny extern . . . . .                    | 61        |
| 5.3.2    | Mapa de navegació . . . . .                 | 62        |
| 5.4      | Capa de domini . . . . .                    | 65        |
| 5.4.1    | Model conceptual . . . . .                  | 65        |
| 5.4.2    | Model de comportament . . . . .             | 66        |
| 5.5      | Capa de dades . . . . .                     | 70        |
| 5.6      | Model de desplegament . . . . .             | 71        |
| 5.7      | Distribució de l'aplicació . . . . .        | 72        |
| <b>6</b> | <b>Desenvolupament</b>                      | <b>73</b> |
| 6.1      | Estructura de l'aplicació . . . . .         | 73        |
| 6.1.1    | Implementació . . . . .                     | 74        |
| 6.2      | <i>Timeline</i> i veure activitat . . . . . | 77        |
| 6.2.1    | Implementació . . . . .                     | 77        |
| 6.3      | Gestió de subscripcions . . . . .           | 82        |
| 6.3.1    | Implementació . . . . .                     | 82        |
| 6.4      | Converses . . . . .                         | 84        |
| 6.4.1    | Implementació . . . . .                     | 84        |
| 6.5      | Perfil . . . . .                            | 88        |
| 6.5.1    | Implementació . . . . .                     | 88        |
| 6.6      | Ampliació converses . . . . .               | 91        |
| 6.6.1    | Implementació . . . . .                     | 91        |
| <b>7</b> | <b>Línies de futur</b>                      | <b>95</b> |



|           |  |            |
|-----------|--|------------|
| <b>8</b>  | <b>Conclusions</b>                             | <b>97</b>  |
| <b>9</b>  | <b>Referencies</b>                             | <b>99</b>  |
| <b>10</b> | <b>Annex</b>                                   | <b>101</b> |
| 10.1      | Glossari . . . . .                             | 101        |
| 10.2      | Històries d'usuari per <i>sprint</i> . . . . . | 103        |
| 10.3      | Llista d'ObjectTypes . . . . .                 | 105        |
| 10.4      | MAX . . . . .                                  | 109        |

## Índex de figures

|    |  |    |
|----|--|----|
| 1  | El procés <i>Scrum</i> . . . . .   | 9  |
| 2  | <i>Sprint Backlog</i> físic de l'equip del SomUPC. . . . .                           | 12 |
| 3  | <i>Sprint Backlog</i> de l'equip del SomUPC utilitzant <i>Trello</i> . . . . .       | 12 |
| 4  | Detall d'una història d'usuari a mig desenvolupar utilitzant <i>Trello</i> . . . . . | 13 |
| 5  | Diagrama de Gantt. . . . .   | 20 |
| 6  | Diagrama de context. . . . .   | 35 |
| 7  | Model conceptual de l'aplicació. . . . .   | 49 |
| 8  | Diagrama de seqüència d'obtenció d'informació. . . . .                               | 50 |
| 9  | Diagrama de seqüència de publicar contingut. . . . .                                 | 50 |
| 10 | Diagrama de seqüència d'esborrat. . . . .  | 50 |
| 11 | Diagrama de seqüència de modificació. . . . .  | 50 |
| 12 | Flux de control del patró MVC. . . . .   | 55 |
| 13 | Comunicació entre el centre de notifikacions i els observadors. . . . .              | 56 |
| 14 | Comunicació vista-controlador mitjançant delegats. . . . .                           | 57 |
| 15 | Logo <i>Activity Streams</i> . . . . .   | 59 |
| 16 | Logo <i>STOMP</i> . . . . .  | 60 |
| 17 | Comparació del disseny i el resultat de la vista . . . . .                           | 61 |
| 18 | Estructura del disseny extern del sistema . . . . .                                  | 62 |
| 19 | Mapa de navegació del <i>timeline</i> . . . . .                                      | 63 |
| 20 | Mapa de navegació de les converses. . . . .  | 63 |
| 21 | Mapa de navegació de les subscripcions. . . . .                                      | 64 |
| 22 | Mapa de navegació del perfil d'usuari. . . . .                                       | 64 |
| 23 | Model conceptual normalitzat. . . . .  | 65 |
| 24 | Diagrama de seqüència d'una operació d'obtenció d'informació. . . . .                | 67 |
| 25 | Diagrama de seqüència d'una operació de publicació d'informació. . . . .             | 68 |
| 26 | Diagrama de seqüència d'una operació d'esborrat. . . . .                             | 69 |
| 27 | Diagrama de seqüència d'una operació de modificació de dades. . . . .                | 70 |
| 28 | Model de desplegament. . . . .   | 71 |
| 29 | Distribució de l'aplicació. . . . .  | 72 |
| 30 | Limit d'ús de la llibreria RestKit. . . . .  | 76 |
| 31 | Resultat de la vista del <i>timeline</i> . . . . .                                   | 78 |
| 32 | Resultat de la vista de veure activitat. . . . .                                     | 78 |
| 33 | Resultat de la vista de publicar comentaris. . . . .                                 | 79 |
| 34 | Resultat de la vista de publicar activitat. . . . .                                  | 79 |
| 35 | Resultat de la vista d'afegir contextos. . . . .                                     | 80 |
| 36 | Resultat de la vista de publicar una activitat amb una imatge. . . . .               | 80 |
| 37 | Adjuntar un fitxer des d'una altra aplicació. . . . .                                | 81 |
| 38 | Resultat de la vista de publicar una activitat amb un fitxer. . . . .                | 81 |
| 39 | Resultat de la vista de subscripcions. . . . .                                       | 83 |
| 40 | Resultat de la vista d'activitats d'un context. . . . .                              | 83 |
| 41 | Resultat de la vista de converses. . . . .   | 85 |
| 42 | Resultat de la vista dels missatges d'una conversa. . . . .                          | 85 |
| 43 | Resultat de la vista d'informació d'una conversa (com a propietari). . . . .         | 86 |
| 44 | Resultat de la vista d'editar la informació d'una conversa. . . . .                  | 86 |
| 45 | Resultat de la vista de nova conversa. . . . .                                       | 87 |

|    |  |    |
|----|--|----|
| 46 | Resultat de la vista de cercar usuaris. . . . .                          | 87 |
| 47 | Resultat de la vista del perfil. . . . .                                 | 89 |
| 48 | Resultat de la vista del perfil en mode d'edició. . . . .                | 89 |
| 49 | Missatge de confirmació abans de tancar sessió. . . . .                  | 90 |
| 50 | Resultat de la vista d'iniciar sessió. . . . .                           | 90 |
| 51 | Llista de converses amb una conversa amb missatges sense llegir. . . . . | 93 |

## Índex de taules

|   |  |    |
|---|--|----|
| 1 | Salari estimat associat a cada rol involucrat al projecte. . . . .   | 22 |
| 2 | Hores de cada rol pels <i>sprints</i> de desenvolupament. . . . .  | 22 |
| 3 | Hores de cada rol pel <i>sprint</i> final. . . . .   | 23 |
| 4 | Cost estimat total dels recursos humans per cada rol. El càlcul seguit està detallat per al cap de projecte a l'equació 2.2. . . . . | 23 |
| 5 | Costos directes estimats. . . . .  | 24 |
| 6 | Cost total estimat. . . . .  | 24 |
| 7 | Histories d'usuari realitzades per <i>sprint</i> . . . . .   | 73 |
| 8 | Significat de les diferents etiquetes. . . . .   | 82 |

# 1 Introducció

Actualment la Universitat Politècnica de Catalunya (UPC) disposa d'una gran quantitat de plataformes per recolzar la docència i la gestió de la universitat. La majoria d'aquestes eines estan dissenyades per assistir als docents i estudiants mitjançant un conjunt d'aplicacions per facilitar la gestió documental, per oferir un canal de comunicació entre el docent i l'estudiant (Atenea o el Racó de la FIB) o simplement un servei molt concret (com pot ser el LEARN-SQL que ofereix un entorn per fer docència de base de dades).

UPCnet[24], que és el client d'aquest projecte (el projecte SomUPC), ha detectat que encara així, hi ha mancances en l'aspecte comunitari. Aquest mètodes no són els ideals per comunicar-se amb la gent jove, i es vol dissenyar una nova plataforma que eviti la dependència al correu electrònic, que tingui les xarxes socials més presents i, que permeti interacció entre els membres de la comunitat. Aquesta mancança està augmentant contínuament degut als canvis d'hàbits i costums de les noves generacions d'estudiants.

El projecte SomUPC consisteix en un portal per integrar totes les plataformes i eines, per a oferir als membres de la UPC un sol punt d'entrar a totes les plataformes. A més, es vol millorar l'aspecte comunitari.

Per aquest motiu, s'ha dissenyat un sistema en el que hi ha un motor d'activitat social i una sèrie d'aplicacions que s'alimenten d'aquest i el nodreixen. Aquest motor d'activitat social rep el nom de MAX i d'ara en endavant l'anomenarem així. A l'annex es pot trobar una explicació més detallada del MAX (secció 10.4).

Aquestes aplicacions poden ser sistemes ja existents, com és el cas d'Atenea, el Racó de la FIB o noves eines com són el portal SomUPC i les aplicacions natives per a dispositius mòbils iOS i Android.

En concret, aquest treball final de grau es concentra en el desenvolupament de l'aplicació iOS per al MAX, que és una de les peces del SomUPC.

UPCnet vol desenvolupar aquesta eina per a la UPC, però també vol que es pugui implantar en qualsevol altra comunitat que ho pugui necessitar, per exemple una empresa. Per tant, el sistema ha de ser molt flexible per a que s'adapti el millor possible a qualsevol organització.



## 2 Gestió del projecte

### 2.1 Metodologia

Pel desenvolupament del SomUPC s'està utilitzant una metodologia àgil. En concret la metodologia *Scrum*. Una metodologia flexible i àgil és una de les claus per a l'èxit del projecte. L'objectiu és el d'aprendre el que l'usuari i el client desitja i el que no desitgen. Es pot donar la situació en la que una funcionalitat inicialment sembli molt útil i fonamental, però després no obtingui una bona recepció per part dels usuaris als que va dirigit el producte.

*Scrum* està basat en iteracions que reben el nom d'*sprints*. Un *sprint* està format per un conjunt de tasques que s'han de completar per aconseguir els seus objectius. Normalment aquestes iteracions duren entre una setmana i un màxim de quatre setmanes. A la figura 1 es pot veure el flux general de la metodologia. A un *sprint*, es desenvolupen completament una o més funcionalitats que es poden posar a producció al acabar la iteració. Els requisits durant els *sprints* són intocables, només es poden modificar entre *sprints*.

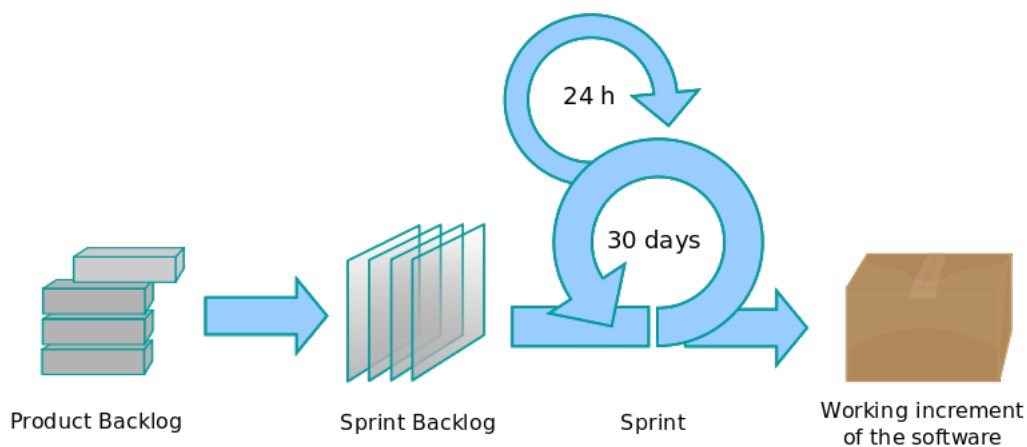


Figura 1: El procés *Scrum* Font: [wikimedia.org](https://www.wikimedia.org/).

Les iteracions curtes permeten obtenir *feedback* continuat dels usuaris i del client. Això permet que el producte canviï continuament en funció de la reacció dels usuaris i del client. Aquesta metodologia accepta i propicia l'aparició de nous requisits, així com la modificació dels requisits existents. Aquests enriqueixen el producte i garanteixen l'èxit del projecte.

Aquesta metodologia permet fer els canvis necessaris per adaptar-la al projecte i a l'equip. En aquest projecte no s'està seguint la metodologia al peu de la lletra, s'han fet algunes

adaptacions per ajustar-la a l'equip. A continuació es detallen alguns conceptes importants i es detalla com s'ha aplicat *Scrum* en aquest projecte.

### 2.1.1 Iteracions - *Sprints*

Com s'ha vist prèviament, el desenvolupament es divideix en *sprints*. L'objectiu principal d'aquesta divisió en iteracions és la d'entregar producte al client o a l'usuari final el més aviat possible. D'aquesta manera es poden detectar els problemes ràpidament i es pot refinar els requisits per a maximitzar les garanties d'èxit. Aquesta filosofia és ideal per a projectes en que s'estiguin desenvolupant productes innovadors, ja que en aquest àmbit la tecnologia avança molt ràpidament i s'ha de permetre pivotar ràpidament entre possibles solucions. Sovint la situació inicial canvia notablement durant el desenvolupament del projecte, i solucions que eren inviables o innecessàries a l'inici del projecte al final no ho són.

A l'inici de l'*sprint* es fa una reunió (anomenada *sprint planning*), en la que participen tots els membres involucrats al projecte (negoci i equip desenvolupador). En aquesta reunió es decideixen les tasques que s'han de realitzar durant el *sprint*.

Prèviament a aquesta reunió s'han definit un conjunt de tasques, que estan al *product backlog* prioritzades en funció del valor que aporten al projecte.

A la reunió d'*sprint planning* s'estableixen les tasques que entren al *sprint*. Aquestes es posen en comú, per a verificar que tots els membres entenen el mateix per cada una de les tasques i l'equip desenvolupador valora les tasques.

A partir d'aquesta valoració i tenint en compte l'ordre de prioritat establert per negoci, es defineix una línia de tall per establir les tasques que entren al *sprint*. Aquestes tasques es col·loquen al *task-board* a la columna de *Sprint backlog*.

L'*sprint* finalitza amb una nova reunió en la que es realitza una demostració del que s'ha fet durant les dues setmanes. A aquesta reunió assisteixen l'equip desenvolupador i negoci. Utilitzant el *task-board*, per cada tasca de la columna de *Done* l'equip desenvolupador demostra que s'ha realitzat i que verifica els criteris de satisfacció.

### 2.1.2 *Daily Scrum*

Cada dia tot l'equip desenvolupador realitza una reunió de deu minuts. Aquesta reunió rep el nom de *Daily Scrum*, també s'anomena *Stand-up meeting* ja que aquesta reunió es



realitza de peu.

En aquesta reunió cada component de l'equip presenta un breu resum de la feina realitzada el dia anterior, i posteriorment diu les tasques que realitzarà durant el dia.

Qualsevol element bloquejant que pugui interferir amb alguna tasca de l'*sprint*, es menciona durant la reunió i es comenta. D'aquesta manera es pot evitar o reduir al mínim les conseqüències. Un element bloquejant és un obstacle que evita que una part del treball es finalitzi.

Aquestes reunions, a més de millorar la coordinació entre els membres, és una font de motivació al generar la sensació d'avançar al veure que les tasques es van finalitzant.

### 2.1.3 Tauler de tasques - *Task board*

La *task board* o tauler de tasques és una visió de l'estat de totes les tasques de l'*sprint*. Aquest panell permet als membres de l'equip visualitzar l'estat del projecte d'aquell moment en un sol cop d'ull. Es pot saber quines tasques s'estan desenvolupant, qui ho està desenvolupant, veure les tasques que estan pendents i les tasques que ja s'han completat.

En aquest projecte s'utilitzen dues versions del panell, que contenen la mateixa informació. S'utilitza una pissarra física amb *post-it* i una versió digital utilitzant l'eina *Trello*<sup>1</sup>. A la figura 2 i 3 es poden veure els dos panells.

El panell està format per quatre columnes:

- ***Sprint backlog***: Tasques preparades per a ser realitzades.
- ***Problems***: Tasques bloquejades.
- ***Doing***: Tasques que cada membre està desenvolupant.
- ***Done***: Tasques ja realitzades.

---

<sup>1</sup>Eina de col·laboració per a organitzar els projectes en taules. En un cop d'ull, *Trello* et diu en que s'està treballant, qui està treballant en què i que està en procés.

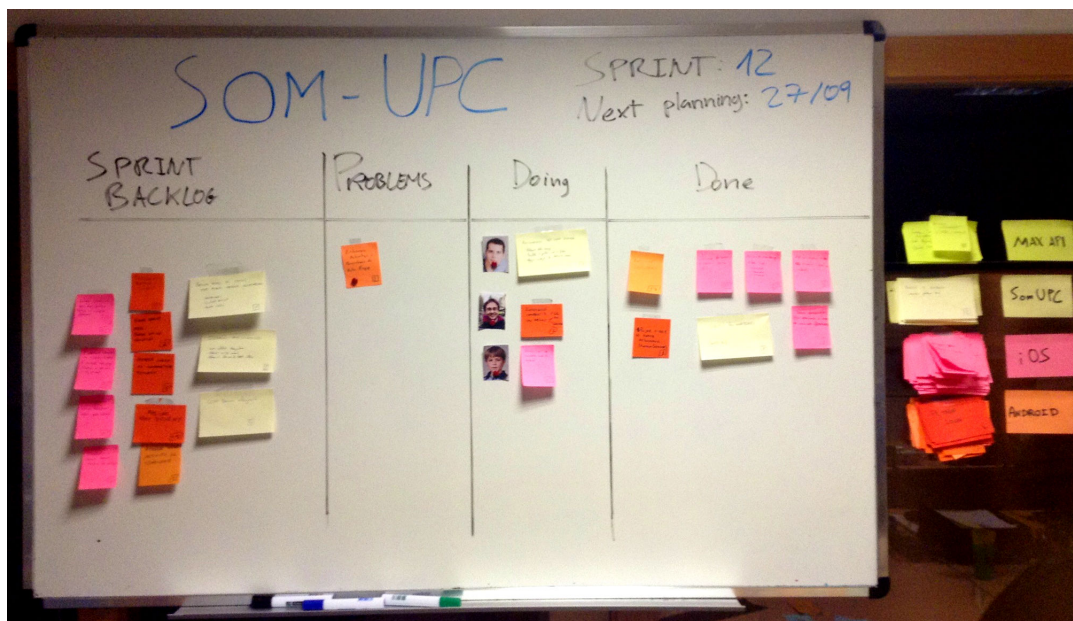


Figura 2: *Sprint Backlog* físic de l'equip del SomUPC.

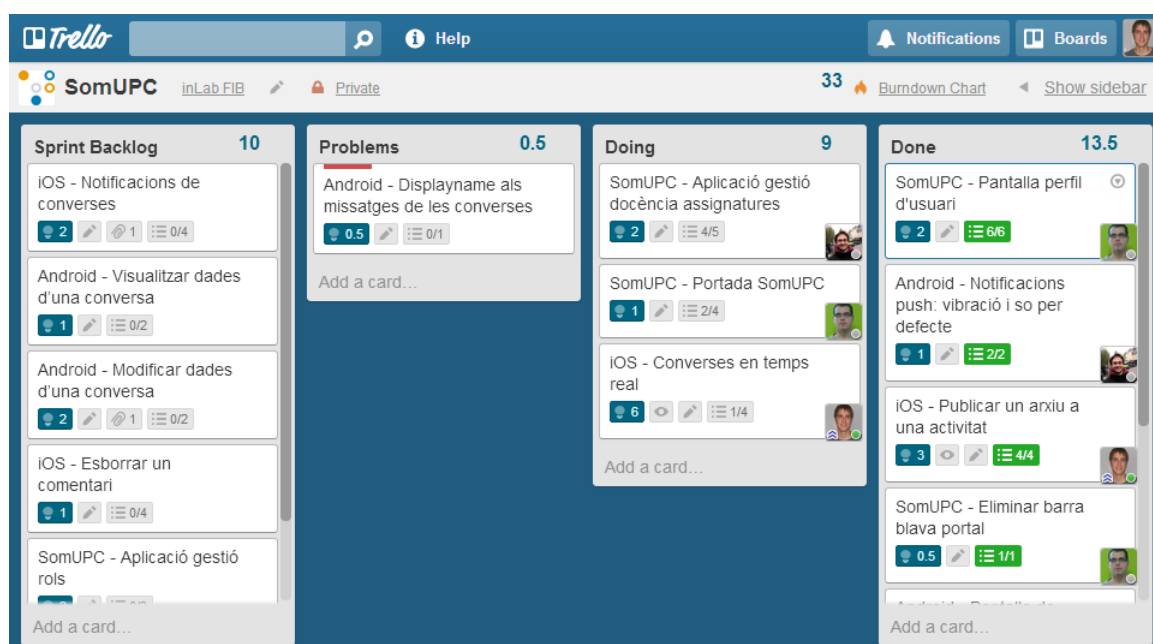


Figura 3: *Sprint Backlog* de l'equip del SomUPC utilitzant *Trello*.

## 2.1.4 Història d'usuari

Els requisits funcionals del projecte s'han definit utilitzant les històries d'usuari. Cada un d'aquests elements defineix una funcionalitat que ha de tenir el sistema.

La història d'usuari consisteix en un parell de línies de text que descriu el requisit des del punt de vista de l'usuari. A més s'inclou un títol descriptiu de la història i un identificador

únic. Aquest identificador no implica cap ordre ni importància, simplement és un codi que s'utilitza per identificar de forma unívoca la història. La història d'usuari també inclou els criteris d'acceptació expressats també des del punt de vista de l'usuari. Tota història d'usuari mostra clarament el motiu i la finalitat per la que s'ha definit.

Es poden veure les històries d'usuari d'aquest projecte a la secció de Històries d'usuari. Durant el desenvolupament, algunes històries d'usuari s'han fragmentat en tasques de menor granularitat, per a facilitar a l'equip desenvolupador la distribució d'aquestes dintre dels *sprints*.

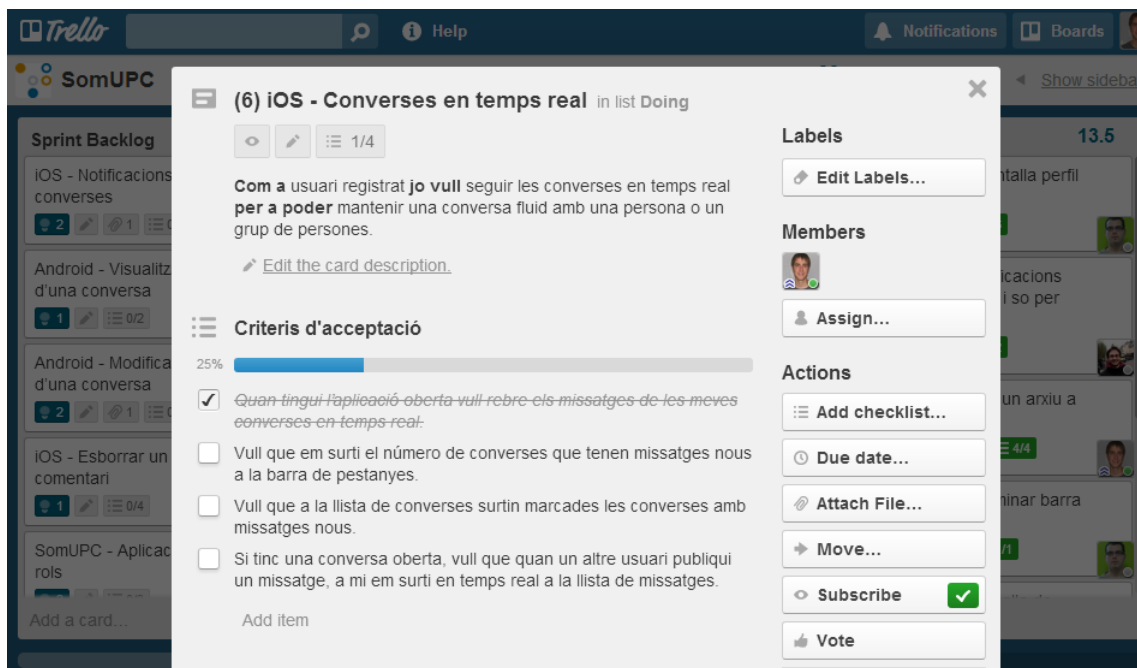


Figura 4: Detall d'una història d'usuari a mig desenvolupar utilitzant *Trello*.

A la figura 4 es pot veure una captura d'una història d'usuari a mig desenvolupar. La captura mostra com es veu una història d'usuari amb la eina utilitzada, *Trello*. Es pot veure la valoració de la història realitzada per l'equip desenvolupador (en aquest cas 6), el títol, el text i els criteris d'acceptació de la història (que en aquest cas són quatre, i un ja està fet).

### 2.1.5 Control de versions

El control de versions en aquest projecte s'ha realitzat amb Git. Git és un sistema distribuït de control de versions. Cada membre de l'equip treballa amb el seu repositori privat a la seva màquina. Quan un membre vol sincronitzar el treball realitzat, envia els canvis al repositori compartit. A aquesta acció se li diu *push*.

El sistema git s'encarrega de realitzar la major part de la barreja dels canvis. Com la majoria d'eines de control de versions, Git no pot barrejar dos versions de dos repositoris que han realitzat canvis a la mateixa part de codi. En aquests casos, Git demana a l'usuari que realitzi els canvis a mà.

El repositori compartit és privat, i el té allotjat el client als seus servidors. Tots els membres de l'equip hi poden accedir des de qualsevol lloc.

## 2.2 Planificació inicial

El projecte es va iniciar al mes de Febrer del 2013 i finalitzarà al mes d'Octubre del 2013, amb una aturada a l'Agost per vacances. S'han establert els *sprints* de dues setmanes, per tant, està previst dur a terme 16 *sprints*. D'aquests, està previst fer 15 de desenvolupament, i un *sprint* final per acabar tasques pendents i resoldre incidències.

Cada *sprint* dura dues setmanes que son 10 dies laborables. Cada dia laborable té 4 hores, però només 2 es dediquen al treball, ja que la dedicació de l'autor al treball final de grau dintre del projecte SomUPC no és completa.

A més es dedicarà mig mes per a la planificació inicial del projecte, i un mes per a la preparació de la documentació del projecte.

Per tant:

$$15 \text{ dies planificació} + 16 \text{ sprints} \cdot 10 \text{ dies} / \text{sprint} + 30 \text{ dies documentació} = 205$$
$$205 \text{ dies} \cdot 2 \text{ h/dia} = 410 \text{ hores} \quad (2.1)$$

Aproximadament es dedicarà el temps que requereix la dedicació del treball final de grau.

### 2.2.1 Recursos temporals

Per desenvolupar aquest treball, el recurs principal és l'autor, que tindrà els rols d'analista, dissenyador, desenvolupador i *tester*. També es disposarà d'assistència per part del client UPCnet sobre el motor MAX, per resoldre dubtes i atendre peticions. Aquest recurs podrà ser utilitzat en cas de necessitat i pot servir per resoldre contratemps de manera ràpida.

No s'han de considerar més recursos materials ja que totes les eines que s'utilitzaran seran lliures, sense cap necessitat de reserva per poder ser utilitzades.

### 2.2.2 Fases del projecte

Per a desenvolupar aquest projecte s'està seguint la metodologia àgil *Scrum*. Per aquest motiu no es poden definir unes fases amb data d'inici i data final, per tant només es pot dividir el projecte en sis fases que s'aniran repartint entre els diferents *sprints* en funció del que demani negoci.

### 2.2.2.1 Estructura de l'aplicació

En aquesta fase s'iniciarà el desenvolupament del projecte creant el *core* de l'aplicació per poder continuar amb les altres fases.

**Requisits:** Cap, aquesta és la primera fase del projecte

**Dificultat estimada:** Mitjana-Alta

**Prioritat:** Molt alta

**Tasques:**

- Crear i estructurar el projecte.
- Preparar l'aplicació perquè es pugui connectar amb el MAX.
- Iniciar sessió amb el sistema d'autenticació de la UPC.

### 2.2.2.2 *Timeline* i veure activitat

Aquesta fase inclourà la funcionalitat a l'aplicació de veure les activitats recents i poder afegir-ne de noves.

**Requisits:** Estructura de l'aplicació finalitzada

**Dificultat estimada:** Mitjana

**Prioritat:** Mitjana

**Tasques:**

- Obtenir les activitats del *timeline*.
- Visualitzar el contingut d'una activitat del *timeline*, el text de l'activitat i els seus comentaris.
- Publicar un nou comentari a una activitat.
- Esborrar un comentari (un comentari propi o a una activitat pròpia).
- Publicar una nova activitat, indicant el text i el context al que es publica.
- Esborrar una activitat (un activitat pròpia).

### 2.2.2.3 Gestió de subscripcions

Aquesta fase incorporarà al projecte la funcionalitat de veure els contextos als que l'usuari està subscrit i que l'usuari pugui gestionar-los.

**Requisits:** Estructura de l'aplicació finalitzada

**Dificultat estimada:** Mitjana

**Prioritat:** Mitjana

**Tasques:**

- Mostrar contextos als que l'usuari està subscrit.

- Visualitzar activitats d'un dels contextos.
- Publicar a un dels contextos.
- Esborrar la subscripció a un dels contextos (si l'usuari te permís per poder-ho fer).
- Afegir una nova subscripció, per fer-ho l'usuari haurà de poder cercar contextos públics.

#### 2.2.2.4 Converses

En aquesta fase s'incorporarà la funcionalitat de veure les converses de l'usuari, poder veure el contingut i afegir nous missatges a una conversa.

**Requisits:** Estructura de l'aplicació finalitzada

**Dificultat estimada:** Mitjana

**Prioritat:** Alta

**Tasques:**

- Mostrar les converses de l'usuari.
- Visualitzar el contingut d'una conversa.
- Afegir un nou missatge a una conversa.
- Veure la informació d'una conversa (nom del grup i participants).
- Sortir o eliminar una conversa (en funció de si l'usuari és el creador del grup, o és una conversa de dues persones).

#### 2.2.2.5 Perfil

Aquesta fase inclourà la funcionalitat de veure el perfil de l'usuari i que aquest pugui ser editat.

**Requisits:** Estructura de l'aplicació finalitzada

**Dificultat estimada:** Baixa

**Prioritat:** Baixa

**Tasques:**

- Mostrar el perfil de l'usuari (nom, *twitter* i activitats de l'usuari)
- Editar el nom i *twitter* de l'usuari
- Afegir una nova activitat al perfil de l'usuari
- Tancar sessió

#### 2.2.2.6 Ampliació converses

Aquesta fase incorporarà al projecte millores al sistema de converses. En concret apor-

tarà temps real a la conversa i notificacions *push*.

**Requisits:** Fase de Converses finalitzada

**Dificultat estimada:** Alta

**Prioritat:** Mitjana

**Tasques:**

- Modificar les converses per rebre missatges en temps real.
- Rebre notificacions *push* a través del sistema APNS<sup>2</sup>.

### 2.2.3 Seguiment de la planificació

Degut a la metodologia que es segueix per desenvolupar el projecte, el seguiment de la planificació és molt senzill de verificar que es compleix.

Només pot haver problemes a l'hora de realitzar els *sprints plannings*. Per exemple, pot passar que per motius d'agenda una reunió no es pugui convocar quan toca, i per tant retardar tota la planificació. Per evitar aquest problema, al principi del projecte s'ha establert que cada dos divendres es farà el *sprint planning* i la demostració, coincidint amb el primer divendres del mes, i el tercer divendres del mes.

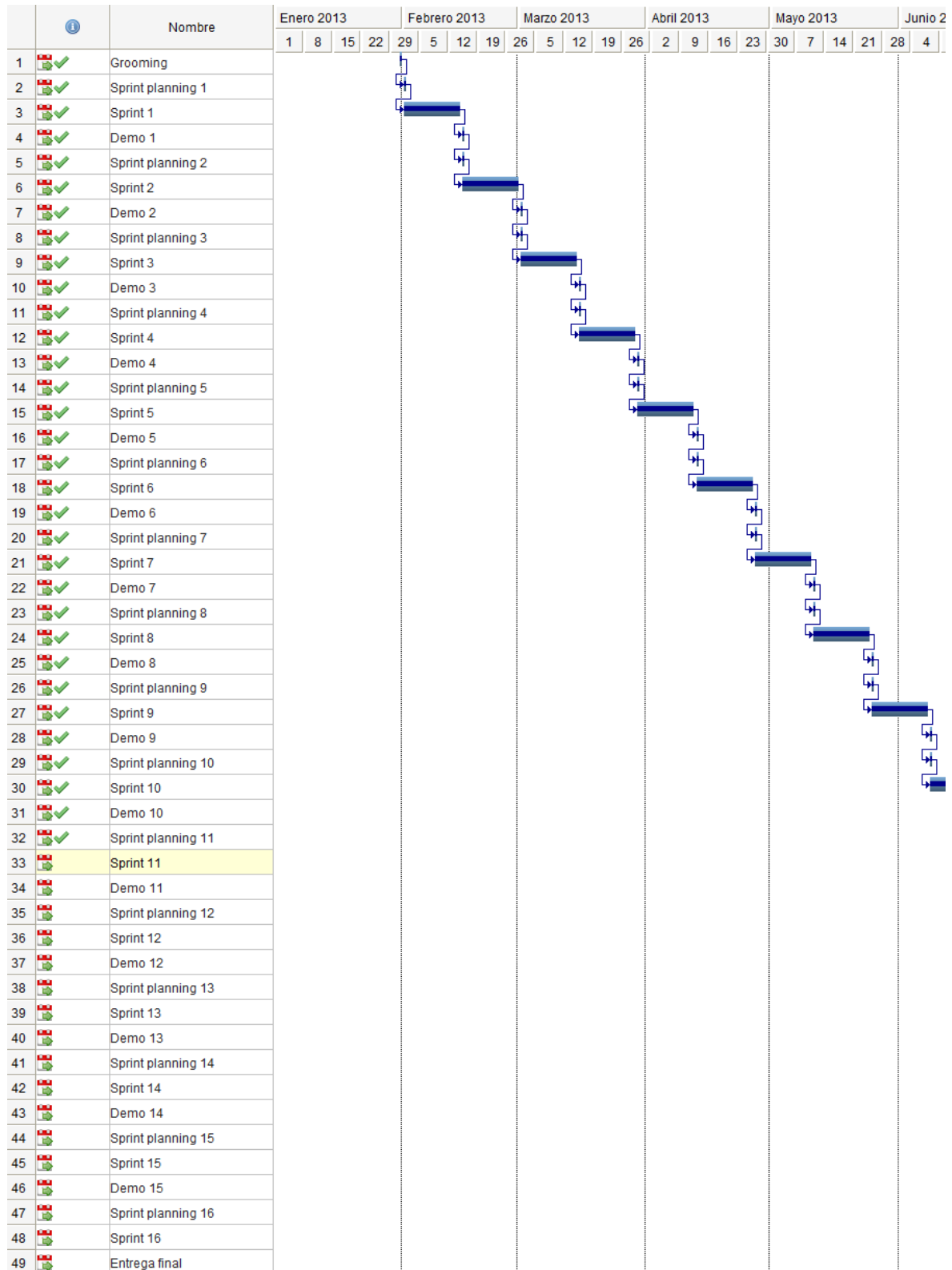
### 2.2.4 Diagrama de Gantt

A la planificació temporal s'han establert tots els *sprints* amb el seu *sprint planning* i la seva demostració. A la figura 5 es pot veure el diagrama de Gantt de la planificació.

---

<sup>2</sup>Apple Push Notification Service





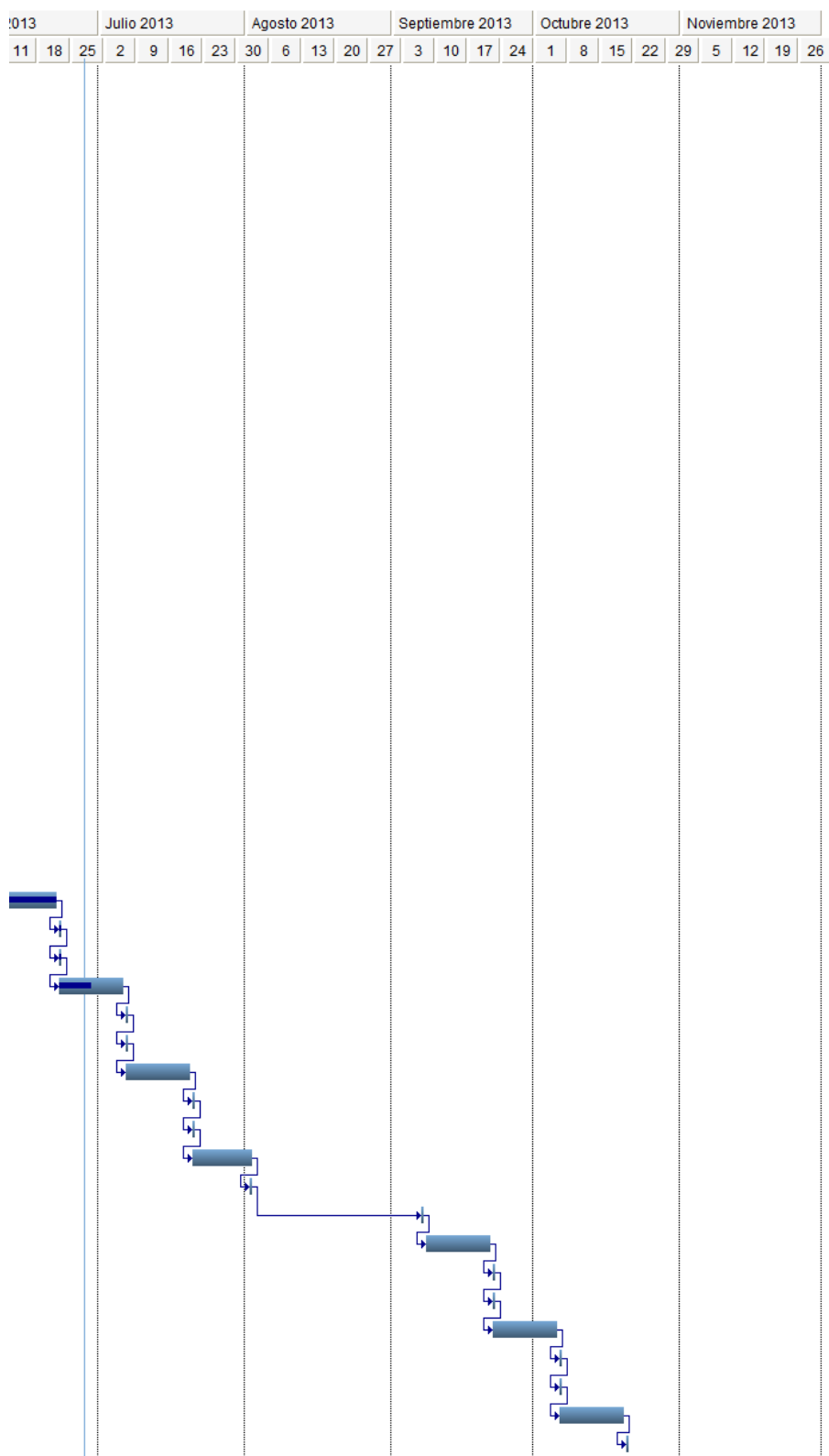


Figura 5: Diagrama de Gantt.

## 2.3 Planificació final

La planificació final d'aquest projecte es correspon amb la planificació inicial establerta a la secció anterior. No hi ha hagut cap desviació de la planificació inicial gracies a l'ús de la metodologia àgil.

S'han realitzat els 16 *sprints* establerts, amb una duració de dues setmanes cada un, com s'havia establert inicialment. A més s'han complert els 15 dies establerts per a la planificació i els 30 dies de documentació final en els que s'ha realitzat aquesta memòria.

El client ha anat ajustant les històries d'usuari que més aportaven al projecte per tal de garantir que entressin dintre dels *sprints* establerts. A l'annex (secció 10.2) es poden veure les històries d'usuari que s'han realitzat a cada un dels *sprints*.

S'han pogut realitzar totes les tasques que es van definir a l'inici del projecte, i alhora s'han realitzat algunes tasques que el client ha definit durant el desenvolupament d'aquest. L'*sprint* final, ha servit per acabar totes les tasques pendents, alhora que s'ha utilitzat per a resoldre els problemes detectats.

## 2.4 Pressupost

### 2.4.1 Recursos humans

El cost principal d'aquest projecte és el de recursos humans. Es poden diferenciar cinc rols que intervindran al projecte: el rol de cap de projecte, el d'analista informàtic, dissenyador, desenvolupador i *tester*. Per a poder calcular el cost dels recursos humans es necessita saber el preu per hora de cada un d'aquests rols, i el temps que es necessitarà de a cada un.

El pressupost es basarà en els preus de la taula 1 que s'han obtingut d'un estudi extret del COEINF<sup>3</sup>[18]

| Rol             | Salari |
|-----------------|--------|
| Cap de projecte | 45 €/h |
| Analista        | 45 €/h |
| Dissenyador     | 30 €/h |
| Programador     | 30 €/h |
| <i>Tester</i>   | 30 €/h |

Taula 1: Salari estimat associat a cada rol involucrat al projecte.

Per a calcular el cost en recursos humans es seguirà la planificació que s'ha realitzat prèviament, en concret es seguirà el diagrama de Gantt. En aquesta planificació s'han previst fer 16 *sprints*, dels quals 15 són de desenvolupament i 1 de proves. Si es considera que per cada un dels 15 *sprints* de desenvolupament, que duren 20 hores, es requereixen les hores indicades a la taula 2. I que a l'últim es requereixen les hores indicades a la taula 3. Es pot estimar que el cost de recursos humans serà de 11.205€, es pot veure a la taula 4 el detall del cost per cada rol.

| Rol             | Hores    |
|-----------------|----------|
| Cap de projecte | 4 hores  |
| Analista        | 3 hores  |
| Dissenyador     | 1 hora   |
| Programador     | 10 hores |
| <i>Tester</i>   | 2 hora   |

Taula 2: Hores de cada rol pels *sprints* de desenvolupament.

$$\text{Cost cap de projecte} = (4 * 15 + 2)\text{hores} * 45\text{€/hora} = 2.790\text{€} \quad (2.2)$$

---

<sup>3</sup>Col·legi Oficial d'Enginyeria en Informàtica de Catalunya

| Rol             | Hores    |
|-----------------|----------|
| Cap de projecte | 2 hores  |
| Analista        | 0 hores  |
| Dissenyador     | 2 hores  |
| Programador     | 3 hores  |
| <i>Tester</i>   | 13 hores |

Taula 3: Hores de cada rol pel *sprint* final.

| Rol             | Cost            |
|-----------------|-----------------|
| Cap de projecte | 2.790 €         |
| Analista        | 2.025 €         |
| Dissenyador     | 510 €           |
| Programador     | 4.590 €         |
| <i>Tester</i>   | 1.290 €         |
| <b>Total</b>    | <b>11.205 €</b> |

Taula 4: Cost estimat total dels recursos humans per cada rol. El càlcul seguit està detallat per al cap de projecte a l'equació 2.2.

#### 2.4.2 Recursos Materials

El cost de recursos materials, al tractar-se d'un projecte en el que es desenvolupa una aplicació iOS, consisteix en un ordinador Mac per a poder desenvolupar. Per aquest projecte s'ha utilitzat un *MacBook Pro* que costa 1.229€<sup>4</sup>.

Per a poder provar l'aplicació en un dispositiu real, també es necessita un iPhone que actualment té un cost de 669€<sup>5</sup>.

A més per a poder instal·lar l'aplicació als dispositius, s'ha de tenir un compte de desenvolupador iOS que té un cost de 80€ anuals<sup>6</sup>.

Al pressupost s'ha de comptabilitzar l'amortització d'aquests costos. Actualment a Espanya el coeficient d'amortització és del 26% per any[3]. La fracció d'any que es dedicarà a treballar en aquest projecte, tenint en compte que la jornada laboral estàndard és de 8 hores diàries i, que en aquest projecte se li dediquen una mitjana de 2 hores diàries durant 160 dies, equival a 40 dies.

$$\text{Amortització} = (1.229 + 669 + 80)\text{€} \cdot 0.26 \cdot 40/365 = 56.36\text{€} \quad (2.3)$$

Per a poder desenvolupar s'utilitzarà software que ve inclòs en el sistema operatiu Mac (per

<sup>4</sup><http://www.apple.com/es/macbook-pro/>

<sup>5</sup><http://www.apple.com/es/iphone/specs.html>

<sup>6</sup><https://developer.apple.com/programs/ios/>

exemple l'entorn de desenvolupament *Xcode*, i software opensource (per exemple git per al control de versions).

### 2.4.3 Costos indirectes

També s'han de contemplar els costos indirectes necessaris per a poder desenvolupar el projecte. Al inLab FIB per als projectes es valoren els costos indirectes a partir d'un percentatge dels costos directes. Actualment aquest percentatge és del 7%.

Aquests costos contemplen l'espai utilitzat, els costos de gestió, les despeses d'aigua i llum, entre altres.

### 2.4.4 Cost total

El cost total estimat del projecte s'obté a partir de la suma dels costos directes i els costos indirectes. Els costos directes són la suma dels recursos materials i els recursos humans.

| <b>Recursos</b> | <b>Cost</b>     |
|-----------------|-----------------|
| Humans          | 11.205 €        |
| Materials       | 56 €            |
| <b>Total</b>    | <b>11.261 €</b> |

Taula 5: Costos directes estimats.

$$\text{Costos indirectes estimats} = 11.261\text{€} \cdot 7/100 = 788.27\text{€} \quad (2.4)$$

El cost total estimat del projecte, tenint en compte els costos directes estimats calculats a la taula 5 i els costos indirectes estimats calculats a l'equació 2.4 és de 12.049 €, es pot veure a la taula 6 el detall dels càlculs.

| <b>Costos</b> | <b>Cost</b>     |
|---------------|-----------------|
| Directes      | 11.261 €        |
| Indirectes    | 788 €           |
| <b>Total</b>  | <b>12.049 €</b> |

Taula 6: Cost total estimat.

### 2.4.5 Control del pressupost

El pressupost d'aquest projecte va directament relacionat amb les hores dedicades al desenvolupament. Un augment del temps dedicat implicaria un augment del cost. Això podria succeir si per exemple es decideixen ampliar el número de *sprints* o la durada d'aquests.

Per evitar un augment del pressupost, l'equip es limitarà a complir la planificació establerta i si el client sol·licita funcionalitats que fan que s'excedeixi la planificació, s'haurà de prescindir d'alguna altra funcionalitat. Serà el client qui haurà de prioritzar les funcionalitats que aporten més valor al producte i per tant entren a dintre de l'abast del projecte.

Si es dona el cas en que el client decideix que hi ha una funcionalitat imprescindible. Però no s'ha previst al principi i és més important desenvolupar-la que una altra funcionalitat prevista. Aleshores es desenvoluparà la nova funcionalitat i no es desenvoluparà la inicial. Aquesta decisió serà presa pel client i consensuada amb l'equip.

### 2.4.6 Viabilitat econòmica

El desenvolupament d'aquest projecte està emmarcat dintre del projecte SomUPC que desenvolupa l'inLab FIB amb el client UPCnet. El SomUPC està finançat pel client amb fons propis per oferir un millor servei a la UPC. A més UPCnet es va comprometre a desenvolupar aquest projecte en una de les edicions dels premis Davyd Luque[23].

UPCnet també té la intenció d'implantar aquests sistemes a empreses externes i, per tant obtindrà un benefici per cada implantació i manteniment.

## **2.5 Sostenibilitat i Responsabilitat Social**

En aquesta secció es tindrà en compte l'impacte medioambiental que produirà el desenvolupament del projecte i els beneficis que aportarà a la societat el producte resultant.

Com s'ha vist anteriorment, els recursos que es necessiten pel desenvolupament del projecte, són majoritàriament recursos humans. El cost de computació i els recursos materials necessaris són molt reduïts i per tant es pot dir que no tindran cap cost mediambiental.

Per altra banda, a la comunitat UPC i en qualsevol altre potencial comunitat, aquesta aplicació pot aportar molts beneficis, especialment pot millorar la forma de comunicar-se entre els membres de la comunitat.

### **2.5.1 Control de la sostenibilitat**

Com s'ha vist anteriorment, aquest projecte al tractar-se d'una aplicació per iOS, no necessita cap control dels efectes mediambientals causats pel desenvolupament ni la posada en producció. Per aquest motiu no s'ha establert cap mecanisme de control mediambiental.



## 2.6 Lleis i regulacions

El projecte SomUPC està subjecte a les lleis de l'estat Espanyol i per tant subjecte a la LOPD<sup>7</sup>. En aquest projecte es guarda informació personal dels usuaris i per tant s'han de seguir tots els passos per complir la LOPD.

Però al sistema d'aquest treball, l'aplicació client del MAX, no es guarda informació personal dels usuaris. Aquesta informació s'obté del servidor cada cop que l'usuari inicia l'aplicació. Per aquest motiu, el sistema desenvolupant en aquest treball queda fora de l'abast de la LOPD.

Al tractar-se d'un client d'una xarxa social, el mal ús d'aquesta eina, pot provocar problemes entre els usuaris. Es pot donar assetjament d'un usuari a un altre usuari, agressions verbals entre usuaris o qualsevol altra situació.

En aquests cas, el responsable de gestionar i solucionar el problema serà el client del projecte, UPCnet. UPCnet conjuntament amb els responsables de la comunitat en que s'estigui utilitzant l'aplicació, prendran les mesures necessàries establertes a la normativa de la comunitat. Per exemple, en el cas de la UPC s'aplicaria la normativa de la UPC[10].

---

<sup>7</sup>Llei de protecció de dades [2]



## 3 Estudi prèvi

### 3.1 Context

En aquest projecte hi han moltes parts interessades involucrades, a continuació es veuran una per una quin rol tenen dintre del SomUPC i quin al projecte del treball final de grau.

#### 3.1.1 UPCnet

UPCnet, al SomUPC, principalment té el rol de client, però també té el rol de desenvolupador d'algunes parts (per exemple el MAX).

Inicialment el SomUPC va sorgir del premi Davyd Luque [23]. En concret a partir de 3 propostes: "UPCesfer@" (edició 2008)<sup>8</sup>, "Directori UPC" (edició 2009)<sup>9</sup> i "U4U - University for you" (edició 2011)<sup>10</sup>. UPCnet quan estava iniciant el projecte va detectar que l'inLab FIB[14] disposava del Racó que tenia moltes similituds amb algunes parts del SomUPC.

En aquest projecte igual que al SomUPC, UPCnet té el rol de client. Com a tal, UPCnet vol obtenir una eina de col·laboració que pugui implantar a la UPC i alhora a qualsevol organització que ho pugui necessitar.

#### 3.1.2 inLab FIB

Com s'ha vist en l'apartat anterior, l'inLab FIB és l'encarregat de desenvolupar una part del SomUPC, en concret el portal SomUPC, l'aplicació nativa per a Android i l'aplicació nativa per iOS. L'equip del SomUPC dintre de l'inLab FIB està format pel director, el co-director i l'autor d'aquest projecte i dos membres més. Al projecte, l'inLab FIB és l'encarregat de desenvolupar el producte.

Com a desenvolupadors del sistema, l'inLab FIB vol finalitzar el projecte en els terminis establerts per concloure amb èxit el projecte i mantenir una bona relació amb el client.

---

<sup>8</sup><http://llocs.upc.edu/premidavydлуque/2008/html/propostes.htm>

<sup>9</sup><http://www.upcnet.es/premi-davydлуque/edicions-anteriors/edicio-2009>

<sup>10</sup><http://www.upcnet.es/premi-davydлуque/edicions-anteriors/2011>

### **3.1.3 Director i co-director**

El director i co-director d'aquest projecte són membres de l'inLab FIB. El co-director és el cap de projecte del SomUPC i el director és el cap de l'equip desenvolupador. Al projecte tindran els mateixos rols que al SomUPC.

Com a membres de l'inLab FIB, el seu objectiu és el mateix que s'ha descrit en l'apartat anterior.

### **3.1.4 Autor**

L'autor d'aquest treball al SomUPC té el rol de desenvolupador. Al projecte farà el rol d'analista, dissenyador, desenvolupador i *tester*. És una de les parts més interessades, ja que el seu objectiu és entregar el treball dins del termini establert per tal de poder finalitzar el Grau que està cursant.

Com a membre de l'inLab FIB, el seu objectiu és el mateix que s'ha descrit anteriorment, a més dels objectius concrets del treball final de grau.

### **3.1.5 Usuari final**

Els usuaris finals són una part interessada molt important ja que el sistema està dirigit a ells. Inicialment els usuaris finals seran els membres de la comunitat UPC, però potencialment ho pot ser qualsevol membre d'una comunitat, com pot ser una empresa.

L'objectiu dels usuaris finals és obtenir una plataforma que els hi faciliti comunicar-se amb altres membres de la seva comunitat, mitjançant la publicació d'activitats i les converses directes amb un altre usuari o un grup.

Si es vol que el projecte tingui èxit, s'ha de desenvolupar un sistema robust, còmode i usable per a que els usuaris finals estiguin contents i facin un bon ús.

## 3.2 Objectius

El resultat del treball serà una aplicació nativa per iOS per visualitzar el contingut del MAX. L'objectiu és desenvolupar-ho en el termini establert per al projecte que finalitza al Octubre del 2013.

Els objectius que s'han de complir són:

- Tenir una aplicació funcional amb tots els requisits establerts a la secció abast.
- Tenir una aplicació amb el disseny institucional de la UPC (guies d'estil de la UPC[8]).
- Tenir una aplicació que compleixi la guia de revisió d'aplicacions d'*Apple (App Review Guidelines [4])*.
- Tenir una aplicació que sigui fàcilment adaptable a altres organitzacions.
- Finalitzar el projecte amb un alt nivell de satisfacció del client UPCnet.

Finalment és important que l'aplicació segueixi uns bons patrons de disseny, per a que sigui re-usable, fàcil de mantenir i fàcil d'ampliar.

### 3.3 Estat de l'art

Aquest projecte és una part del SomUPC i, per tant, per analitzar l'estat de l'art es farà de tot el projecte SomUPC.

#### 3.3.1 Entorn de col·laboració

Actualment hi ha moltes eines disponibles a la xarxa per treballar en col·laboració en una comunitat. Totes permeten comunicar informació entre un grup determinat de persones. Al mercat es poden trobar les següents aplicacions que ofereixen requisits similars als que demana el client:

##### 3.3.1.1 Comunicació entre usuaris

Hi ha moltes plataformes que ofereixen als usuaris la possibilitat de comunicar-se. Algunes ho tenen com a propòsit principal, com és el cas de *Whatsapp*[25] o *Hangouts*[13] de *Google*, mentre que altres aplicacions ho tenen com a part del seu producte, com és el cas de *Facebook*[9] o *Twitter*[22], però en aquest cas ofereixen més funcionalitats.

Per altra banda també hi ha plataformes que tenen integrat un sistema de comunicació per facilitar el treball en col·laboració, com és el cas de *Google Drive*[11] o *Cacoo*[6], però la part de comunicació només serveix per millorar l'experiència de l'usuari, ja que l'aplicació té un altre propòsit que no és el de la comunicació.

##### 3.3.1.2 Publicar activitats

Una altra funcionalitat sol·licitada és poder compartir informació amb els altres usuaris mitjançant la publicació d'activitats. Aquestes activitats inicialment només consten d'un text, però es pot ampliar a continguts multimèdia o altres tipus de fitxers.

Aquesta funcionalitat és la base de moltes xarxes socials, *Twitter* per exemple ho permet fer mitjançant la publicació de *tweets*. *Facebook* també ho permet fer mitjançant la publicació d'estats al mur personal. *Google +*[12] ho permet fer mitjançant la publicació de novetats amb un sistema similar al de *Facebook*. D'aquesta manera un altre usuari pot veure les activitats publicades accedint al mur de l'usuari.

A més, es demana que es pugui publicar una activitat en un context concret (és a dir, per a una agrupació de persones concreta). *Facebook* permet publicar activitats a un grup o a una pàgina de *Facebook* i *Google +* permet publicar novetats a un cercle concret. En tots

dos casos només podran veure les activitats publicades aquells usuaris que formin part del grup/cercle o siguin fans de la pàgina.

Un usuari ha de poder publicar un comentari de l'activitat. Això ho permeten les tres xarxes socials, però de dues maneres diferents. *Twitter* permet fer *retweet*, publicar un nou *tweet* com a resposta o fer una menció en un nou *tweet*. En canvi, *Facebook* i *Google +* permeten publicar un comentari directament a l'activitat. Aquest últim mètode és el que s'ha demanat per aquest projecte.

### 3.3.1.3 Subscripció

Els usuaris a les xarxes socials poden accedir als diferents murs dels usuaris per poder veure les novetats, però gairebé totes les xarxes socials tenen una pàgina on es mostren les activitats dels usuaris relacionats agrupats, sovint ordenats cronològicament o per rellevància.

*Facebook* disposa de la secció "Canal de notícies" en que mostra les activitats dels amics, dels grups i de les pàgines a les que l'usuari es fan (per exemple, nous estats o noves fotografies), *Google +* disposa d'una secció similar que es diu "Novetats". *Twitter* també disposa del *timeline*, en el que es poden veure tots els *tweets* dels usuaris als que es segueixen.

El requisit d'aquest projecte és molt similar a aquestes implementacions, però ha de ser més genèric ja que ha de permetre crear grups d'usuaris molt fàcilment (anomenats contexts). Els usuaris s'han de poder subscriure als contexts públics.

D'aquesta manera el producte final es podrà adaptar a qualsevol organització (educativa o empresarial). Per exemple, en el cas de la UPC, cada departament podria tenir el seu context, però també els estudiants de la FIB de Grau podrien tenir un context propi.

L'objectiu és que tot aquell grup d'usuaris que hagi de compartir una activitat pugui tenir un context associat i crear una sub-comunitat (dintre la comunitat).

### 3.3.2 Innovació

Com s'ha vist a l'apartat anterior, ja existeixen sistemes a la xarxa que permeten fer pràcticament tots els requisits del projecte. Aquests són els problemes que s'han detectat als sistemes existents:

- Hi ha moltes plataformes, però no hi ha una que englobi tots els requisits que vol el client en un mateix sistema.

- Per poder utilitzar aquests sistemes en una comunitat, tots els membres haurien d'estar registrats. En molts casos els usuaris ja disposarien d'un usuari en alguna de les plataformes, però potser no desitjarien barrejar l'àmbit professional/acadèmic amb el personal.
- Amb les eines actuals seria impossible englobar tots els usuaris d'una comunitat (com és la UPC) i fer que els usuaris poguessin cercar un usuari de la comunitat entre tots els usuaris del sistema.
- Algunes de les plataformes actuals només estan disponibles en un dels entorns que requereix el client (només web o només app nativa) i no en tots dos entorns.

Durant la recerca de l'estat de l'art s'ha trobat una empresa catalana que ofereix un producte que disposa de molts dels requisits que demana el client, el producte es diu *Zyncro*[26] de l'empresa *Zyncro Tech*. L'equip ha provat la versió gratuïta del producte i les conclusions que s'han extret són les següents:

- No hi ha la possibilitat d'integrar el sistema a les plataformes web de la comunitat/empresa.
- Excepte a les versions *Enterprise* i *Business Communities* les dades estan als servidors d'aquest sistema i no als de l'empresa.
- Disposa d'una aplicació per dispositius mòbils, però és una única aplicació per totes les comunitats. No és possible que cada empresa tingui la seva pròpia aplicació, ni que ho integri a dintre d'una altra que ja estigui al mercat.

Tenint en compte les conclusions extretes, el producte *Zyncro* no s'ajusta als requisits del client.

Per aquest motiu aquest projecte té com objectiu desenvolupar un entorn que faciliti la col·laboració en xarxa dels membres d'una organització. Aquest treball final de grau, en concret es centra en l'app iOS d'aquest sistema.



### 3.4 Abast

Com s'ha dit anteriorment, aquest projecte es centra en el desenvolupament d'una aplicació nativa per a dispositius iOS (iPhone, iPod i iPad) que consisteix en un client de la plataforma MAX. Aquesta aplicació ha d'incloure un conjunt de funcionalitats per tal de millorar la comunicació entre els usuaris d'una comunitat. Per fer-ho oferirà al usuari la possibilitat de crear activitats en context concret i comunicar-se amb un sistema de missatgeria privada amb un altre usuari o un grup d'usuaris.

La gran dificultat que pot aparèixer durant el desenvolupament del projecte és tenir algun problema amb el MAX. Com que el MAX està desenvolupat per UPCnet, hi ha comunicació directe amb l'equip desenvolupador, per tal de poder solucionar qualsevol dubte o problema.

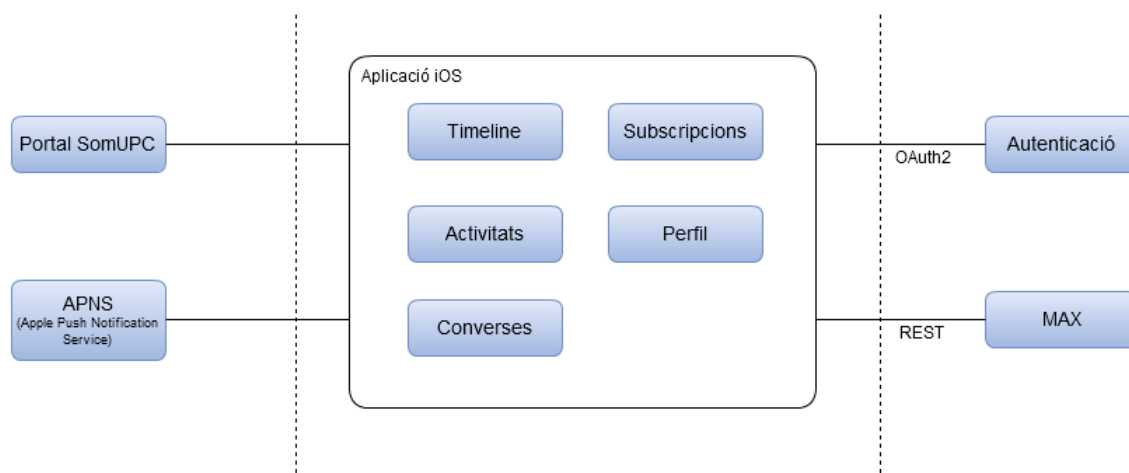


Figura 6: Diagrama de context.

El projecte s'emmarca dintre del SomUPC, per tant és molt important delimitar les fronteres del sistema. El sistema a desenvolupar es tracta exclusivament de l'aplicació iOS, aquesta utilitzarà serveis externs, però aquests no formen part de l'abast. Al diagrama de context de la figura 6 es poden veure clarament les barreres del projecte que delimiten l'abast.

L'aplicació utilitza el MAX com a font de l'activitat social, el portal SomUPC per a obtenir les fotografies dels usuaris, el APNS<sup>11</sup> d'Apple per a les notificacions *push* i el servidor d'autenticació de la UPC per a l'autenticació dels usuaris.

<sup>11</sup>Apple Push Notification Service

### 3.5 Mètodes de validació

Al seguir la metodologia *Scrum*, no es necessita definir criteris d'acceptació de tot el projecte sencer, ja que la pròpia dinàmica ofereix un fort control del producte per part de negoci.

Un dels millors aspectes d'aquesta metodologia és la gran cohesió que crea entre tots els membres involucrats al projecte. En general sempre hi ha una mateixa visió del producte que s'està creant, i negoci pot anar perfilant el producte durant el transcurs dels *sprints*.

No es defineixen criteris d'acceptació de tot el producte, però sí que es defineixen criteris d'acceptació de les diferents tasques. Gràcies a que les tasques tenen una granularitat molt petita, els criteris d'acceptació poden ser molt precisos i concisos.

Com s'ha vist anteriorment, els encarregats de validar els criteris d'acceptació de les tasques són els membres de negoci, durant les reunions de demostració.

### 3.6 Riscs

En aquest projecte s'han de tenir en compte els riscos involucrats al procés de desenvolupament per detectar-los el més aviat possible. Per aquest projecte s'han detectat aquests riscos:

- **Producte no desitjat pel client.** Al tractar-se d'un producte destinat a un client, hi ha la possibilitat que un cop finalitzat el projecte, el client rebi un producte que no és el que esperava obtenir. Per evitar aquest risc, com s'ha indicat en els apartats anteriors, es segueix la metodologia *Scrum* i per tant el client obté un *feedback* continuat durant els *sprints*. Si en algun moment el client detecta que el projecte no és el que espera, el client pot comunicar-ho al següent *sprint* i re-orientar el projecte.
- **Control del temps.** Un risc molt important és excedir el termini de temps establert per a aquest projecte. Per evitar aquest risc, es farà un control exhaustiu per tal de que es compleixi la planificació establerta.
- **Producte poc eficient.** Al tractar-se d'un producte final, destinat a tot tipus d'usuari, el producte resultant ha de tenir un bon rendiment en velocitat d'ús. Per evitar aquest risc, durant el desenvolupament del projecte es tindrà molt en compte la velocitat d'execució.
- **Ús inadequat.** Com que al producte final podran accedir tot tipus d'usuari, hi ha la possibilitat que algun usuari faci un ús inadequat del producte. Per evitar aquest risc, durant el desenvolupament es tindrà en compte la seguretat de l'aplicació per evitar usos inadequats.

### 3.7 Definició de termes i acrònims

En aquesta memòria s'utilitzaran una sèrie de termes freqüentment. Per a facilitar la comprensió d'aquesta a continuació estan definits els termes més habituals. Al glossari (a la secció 10.1) hi han més termes definits.

**Activitat:** al projecte SomUPC, és un conjunt d'informació que vol publicar un usuari a un context (o un grups de contexts) concrets. Pot ser un text, una imatge o un fitxer qualsevol.

**APNS - Apple Push Notification Service:** és un servei creat per *Apple Inc*, que va ser llançat amb el iOS 3.0 el 17 de juny de 2009. El servei utilitza la tecnologia *push* per enviar notificacions dels servidors d'aplicacions de tercers als dispositius *iOS*, les notificacions poden incloure logotips, cançons o alarmes de text personalitzades.

**Context:** al projecte SomUPC, és un conjunt de persones que comparteixen alguna característica en comú, per exemple tots els alumnes d'una assignatura o tots els membres d'una facultat.

**Negoci:** client del producte. Negoci està representat pel *product owner* dintre de *Scrum*.

**Notificació push:** descriu un estil de comunicacions a Internet on la petició d'una transacció s'origina en el servidor. Al contrari a la tecnologia pull, on la petició és originada en el client-servidor.

**Scrum:** marc de treball per a la gestió de projectes que defineix un conjunt de pràctiques, on cada persona participant assumeix un rol (*Scrum master*, *Product owner* i equip de desenvolupament), fet que permet adaptar-se a les necessitats i preferències de cada equip o organització.

**Subscripció:** acció que realitza l'usuari per rebre al seu *timeline* totes les activitats d'un context.

**Timeline:** espai on l'usuari pot veure totes les activitats que s'han publicat al sistema als contexts on està subscrit.

## 4 Especificació

### 4.1 Requisits funcionals

En aquest projecte s'han definit els requisits funcionals seguint el model d'històries d'usuari, ja que s'ha seguit la metodologia *scrum*. Cada una de les històries d'usuari descriu una funcionalitat del sistema.

#### 4.1.1 Històries d'usuari

La metodologia *scrum* utilitza les històries d'usuari per definir les diferents funcionalitats del sistema. Aquestes històries d'usuari estan definides pel client i posades en comú amb tot l'equip als *sprint plannings*.

Cada historia d'usuari esta composta per un títol que les identifica, el text de la història, un número d'història d'usuari (que no implica cap ordre ni prioritat), una xifra de prioritat i els criteris d'acceptació.

Els criteris d'acceptació son els criteris que ha de complir la funcionalitat a desenvolupar, per a que negoci accepti la funcionalitat com a bona.

|  |    |
|--|----|
| <b>1. Iniciar sessió</b>   |    |
| <b>Com a</b> usuari <b>jo vull</b> poder iniciar sessió amb el meu usuari <b>per a poder</b> accedir a la meva informació personal.  |    |
| <b>Prioritat:</b>  | 80 |
| <b>Criteris d'acceptació:</b>  |    |
| <ul style="list-style-type: none"><li>• Vull emplenar un formulari amb les meves credencials per iniciar sessió.</li><li>• No he de poder entrar amb unes credencials invalides.</li></ul> |    |

## 2. Veure *timeline*

**Com a** usuari registrat **jo vull** veure les activitats del meu *timeline* **per a poder** veure les activitats publicades als espais que estic subscrit.

**Prioritat:** 75

### Criteris d'acceptació:

- Vull veure totes les activitats del meu *timeline*.
- Per cada activitat vull veure el començament del contingut, l'autor (el nom i la seva imatge), la data de publicació i el número de comentaris.
- Vull poder clicar a sobre per poder veure el detall de l'activitat.

## 3. Veure activitat

**Com a** usuari registrat **jo vull** accedir a la informació d'una activitat **per a poder** veure el contingut que s'ha publicat, l'autor que ho ha fet i el moment en que ho ha fet.

**Prioritat:** 60

### Criteris d'acceptació:

- Vull veure el contingut complet de l'activitat.
- Vull veure el nom i la imatge del seu autor.
- Vull veure la data de publicació de l'activitat.

## 4. Veure comentaris d'una activitat

**Com a** usuari registrat **jo vull** poder veure els comentaris d'una activitat **per a poder** saber la opinió o els comentaris que han publicat els altres usuaris de l'activitat.

**Prioritat:** 60

### Criteris d'acceptació:

- Vull veure tots els comentaris que s'han publicat a una activitat.
- De cada comentari vull veure el contingut del comentari, l'autor (nom i imatge) i la data de publicació del comentari.

## 5. Crear una activitat

**Com a** usuari registrat **jo vull** publicar una activitat a un espai al que tinc permisos d'escriptura **per a poder** compartir una informació amb els altres usuaris.

**Prioritat:** 65

### Criteris d'acceptació:

- Vull escriure a un camp el contingut de l'activitat.
- Vull seleccionar dels contextos als que puc escriure a quin/quins publicar l'activitat.
- No he de poder publicar a un context on no tingui permís d'escriptura.
- Quan guardi, s'ha de publicar l'activitat.

## 6. Afegir un comentari

**Com a** usuari registrat **jo vull** afegir un comentari a una activitat **per a poder** fer saber als altres usuaris una opinió o comentari.

**Prioritat:** 50

### Criteris d'acceptació:

- Vull escriure a un camp el meu comentari.
- Quan guardi s'ha de publicar el comentari a l'activitat.

## 7. Veure subscripcions

**Com a** usuari registrat **jo vull** veure totes les meves subscripcions **per a poder** revisar els espais als que estic subscrit.

**Prioritat:** 70

### Criteris d'acceptació:

- Vull veure totes les meves subscripcions agrupades per tipologia (Assignatures, Institucionals, Comunitats o altres).
- Si selecciono un context, vull veure les activitats d'aquest context.

## 8. Veure activitats d'una subscripció

**Com a** usuari registrat **jo vull** veure les activitats d'un espai al que estic subscrit **per a poder** veure totes les activitats que han generat els usuaris de l'espai.

**Prioritat:** 65

### Criteris d'acceptació:

- Vull veure totes les activitats publicades a un context.
- Per cada activitat vull veure el començament del contingut, l'autor (el nom i la seva imatge), la data de publicació i el número de comentaris.
- Vull poder clicar a sobre per poder veure el detall de l'activitat.

## 9. Afegir una subscripció

**Com a** usuari registrat **jo vull** afegir noves subscripcions **per a poder** veure les activitats publicades a un espai del que ara no estic subscrit.

**Prioritat:** 65

### Criteris d'acceptació:

- Vull afegir una nova subscripció a un context.
- No he de poder subscriurem a un context que no sigui públic.
- Se m'ha de confirmar si realment vull subscriurem al context.
- Si accepto, s'ha de crear la subscripció al context.
- Si cancel·lo, no s'ha de crear la subscripció.

## 10. Cercar contextos públics

**Com a** usuari registrat **jo vull** cercar un context públic **per a poder** subscriurem al context.

**Prioritat:** 65

### Criteris d'acceptació:

- Vull tenir un camp a on introduir el nom del context a cercar.
- He de veure els contextos que compleixen la cerca.



## 11. Veure perfil usuari

**Com a** usuari registrat **jo vull** veure el meu perfil d'usuari **per a poder** veure quines dades meves té el sistema, i revisar que les dades publicades són correctes.

**Prioritat:** 70

### Criteris d'acceptació:

- Vull veure el meu nom.
- Vull veure el meu nom d'usuari.
- Vull veure el meu *twitter*.
- Vull veure la meva imatge.

## 12. Veure activitats d'un usuari

**Com a** usuari registrat **jo vull** veure les activitats d'un usuari **per a poder** veure les activitats que aquest ha generat al sistema.

**Prioritat:** 50

### Criteris d'acceptació:

- Vull veure totes les activitats que he publicat.
- Per cada activitat vull veure el començament del contingut, l'autor (el meu nom i la meva imatge), la data de publicació i el número de comentaris.
- Vull poder clicar a sobre per poder veure el detall de l'activitat.

## 13. Modificar dades perfil usuari

**Com a** usuari registrat **jo vull** modificar les meves dades d'usuari **per a poder** actualitzar la informació que està publicada al sistema.

**Prioritat:** 60

### Criteris d'acceptació:

- Vull tenir un camp per modificar el meu nom, i que inicialment posi el meu nom antic.
- Vull tenir un camp per modificar el meu *twitter*, i que inicialment posi el meu *twitter* antic.
- Si guardo, vull que el meu perfil es modifiqui amb les noves dades.

## 14. Veure converses usuari

**Com a** usuari registrat **jo vull** veure totes les converses a les que participo **per a poder** triar una conversa i veure els missatges de la conversa.

**Prioritat:** 80

### Criteris d'acceptació:

- Vull veure totes les meves converses.
- Per cada conversa vull veure el començament del últim missatge, el nom del grup, la data del últim missatge, el número de missatges i la imatge de la conversa (si és una conversa de dos, la imatge de l'altre participant, si és de més una imatge fixe).
- Vull poder clicar a sobre per poder veure els missatges de la conversa.

## 15. Veure missatges d'una conversa

**Com a** usuari registrat **jo vull** veure els missatges d'una conversa **per a poder** fer el seguiment de la conversa, llegint els missatges nous o els missatges ja llegits.

**Prioritat:** 80

### Criteris d'acceptació:

- Vull veure tots els missatges de la conversa.
- Els missatges els vull veure en format de xat.
- Per a cada missatge vull veure el text del missatge i la data de quan es va enviar.
- Per als missatges que no sigui meus, vull veure el nom de l'autor del missatge.

## 16. Afegir un nou missatge a una conversa

**Com a** usuari registrat **jo vull** afegir un nou missatge a una conversa **per a poder** comunicar un missatge als altres usuaris de la conversa.

**Prioritat:** 70

### Criteris d'acceptació:

- Vull tenir un camp a on poder posar el text d'un nou missatge.
- Si no he introduït cap missatge, no he de poder enviar.
- Quan envii el missatge, aquest s'ha d'enviar a la resta d'usuaris.
- Quan envii el missatge, el botó d'enviar s'ha de desactivar i s'ha de buidar el camp de text.

## 17. Visualitzar dades d'una conversa

**Com a** usuari registrat **jo vull** visualitzar les dades d'una conversa **per a poder** veure el nom del grup i els membres que hi participen.

**Prioritat:** 60

### Criteris d'acceptació:

- Vull veure el nom de la conversa.
- Vull veure el nom i foto de tots els participants de la conversa.
- A la llista vull veure destacat el creador de la conversa.

## 18. Modificar dades d'una conversa

**Com a** usuari registrat propietari de la conversa **jo vull** modificar les dades de la conversa **per a poder** canviar el nom del grup i afegir o treure usuaris de la conversa.

**Prioritat:** 50

### Criteris d'acceptació:

- Vull tenir un camp per modificar el nom de la conversa, amb el nom antic inicialment.
- Vull poder afegir nous participants a la conversa.
- Vull poder treure participants de la conversa.
- Se m'ha de confirmar abans de treure o posar una persona a la conversa.
- Si accepto, s'ha de afegir/treure a la persona.
- Si cancel·lo, no s'ha de afegir/treure a la persona.
- No he de poder modificar la conversa si no soc el creador d'aquesta.

## 19. Crear una conversa

**Com a** usuari registrat **jo vull** iniciar una nova conversa **per a poder** iniciar una conversa amb un conjunt d'usuaris.

**Prioritat:** 60

### Criteris d'acceptació:

- Vull tenir un camp per afegir els participants.
- Vull tenir un camp per introduir el missatge.
- Si creo la conversa i hi han més de dos participants, se m'ha de preguntar el nom del grup.

## 20. Notificacions de converses

**Com a** usuari registrat **jo vull** rebre notificacions al dispositiu mòbil dels nous missatges **per a poder** assabentar-me que un altre usuari ha enviat un missatge a una conversa en la que participo.

**Prioritat:** 50

### Criteris d'acceptació:

- Quan algun usuari hagi enviat un missatge a una conversa meva, vull que se'm notifi-qui d'aquest missatge.
- No vull rebre notificacions quan l'aplicació estigui oberta.

## 21. Converses en temps real

**Com a** usuari registrat **jo vull** seguir les converses en temps real **per a poder** mantenir una conversa fluida amb una persona o un grup de persones.

**Prioritat:** 50

### Criteris d'acceptació:

- Quan tingui l'aplicació oberta vull rebre els missatges de les meves converses en temps real.
- Vull que em surti el número de converses que tenen missatges nous a la barra de pestanyes.
- Vull que a la llista de converses surtin marcades les converses amb missatges nous.
- Si tinc una conversa oberta, vull que quan un altre usuari publiqui un missatge, a mi em surti en temps real a la llista de missatges.

## 22. Esborrar una activitat

**Com a** usuari registrat **jo vull** esborrar una activitat que he publicat **per a poder** evitar que els altres usuaris vegin una activitat que no vull que puguin veure.

**Prioritat:** 35

### Criteris d'acceptació:

- Vull esborrar una activitat lliscant la cel·la de l'activitat cap a un costat.
- Abans d'esborrar-ho, vull que se'm demani confirmació de l'eliminació.
- Si accepto, s'ha d'esborrar l'activitat.
- Si cancel·lo, no s'ha d'esborrar l'activitat.

## 23. Esborrar un comentari

**Com a** usuari registrat **jo vull** esborrar un comentari meu o un comentari d'una activitat que he publicat **jo per a poder** evitar que els usuaris vegin un comentari que no vull que puguin veure.

**Prioritat:** 30

### Criteris d'acceptació:

- Vull esborrar un comentari lliscant la cel·la del comentari cap a un costat.
- Abans d'esborrar-ho, vull que se'm demani confirmació de l'eliminació.
- Si accepto, s'ha d'esborrar el comentari.
- Si cancel·lo, no s'ha d'esborrar el comentari.

## 24. Esborrar una subscripció

**Com a** usuari registrat **jo vull** esborrar una subscripció **per a poder** deixar de veure les seves activitats al meu *timeline*.

**Prioritat:** 40

### Criteris d'acceptació:

- Vull esborrar una subscripció lliscant la cel·la de la subscripció cap a un costat.
- Abans d'esborrar-ho, vull que se'm demani confirmació de l'eliminació.
- Si accepto, s'ha d'esborrar la subscripció.
- Si cancel·lo, no s'ha d'esborrar la subscripció.

## 25. Sortir/Esborrar una conversa

**Com a** usuari registrat **jo vull** esborrar/sortir una conversa **per a poder** deixar de rebre els missatges d'una conversa que no vull seguir més.

**Prioritat:** 40

### Criteris d'acceptació:

- Vull esborrar/sortir una conversa lliscant la cel·la de la conversa cap a un costat.
- Abans d'esborrar-ho, vull que se'm demani confirmació de l'eliminació.
- Si accepto, s'ha d'esborrar/sortir de la conversa.
- Si cancel·lo, no s'ha d'esborrar/sortir de la conversa.
- Només puc esborrar una conversa si sóc el propietari.
- Només puc sortir d'una conversa si no sóc el propietari.

## 26. Tancar sessió

**Com a** usuari registrat **jo vull** tancar la sessió **per a poder** canviar d'usuari.

**Prioritat:** 40

### Criteris d'acceptació:

- Vull tenir un botó al perfil d'usuari per tancar la sessió.
- Abans de tancar la sessió se m'ha de demanar una confirmació.
- Si accepto, s'ha de tancar la meva sessió a l'aplicació.
- Si cancel·lo, no s'ha de tancar la meva sessió a l'aplicació.

## 27. Publicar un arxiu a una activitat

**Com a** usuari registrat **jo vull** afegir una imatge o un fitxer a una activitat **per a poder** compartir-la amb els altres usuaris de l'espai.

**Prioritat:** 50

### Criteris d'acceptació:

- Vull tenir un botó per afegir una imatge (de la galeria o de la càmera) a l'activitat que estic publicant.
- Vull que des de les altres aplicacions pugui obrir fitxers amb aquesta aplicació, i quan ho faci, que se'm posi com a fitxer adjunt.
- No he de poder publicar una activitat amb un fitxer al meu context personal.
- No he de poder publicar una activitat amb un fitxer a un context que no és una comunitat.

## 4.1.2 Model conceptual

Com a resultat d'aplicar la metodologia *scrum*, l'especificació del model conceptual s'ha anat fent incrementalment durant el desenvolupament del projecte. A cada *sprint* s'ha anat ampliant i perfeccionant el model conceptual.

A continuació es pot veure el model conceptual resultant del projecte al finalitzar els *sprints* (figura 7).

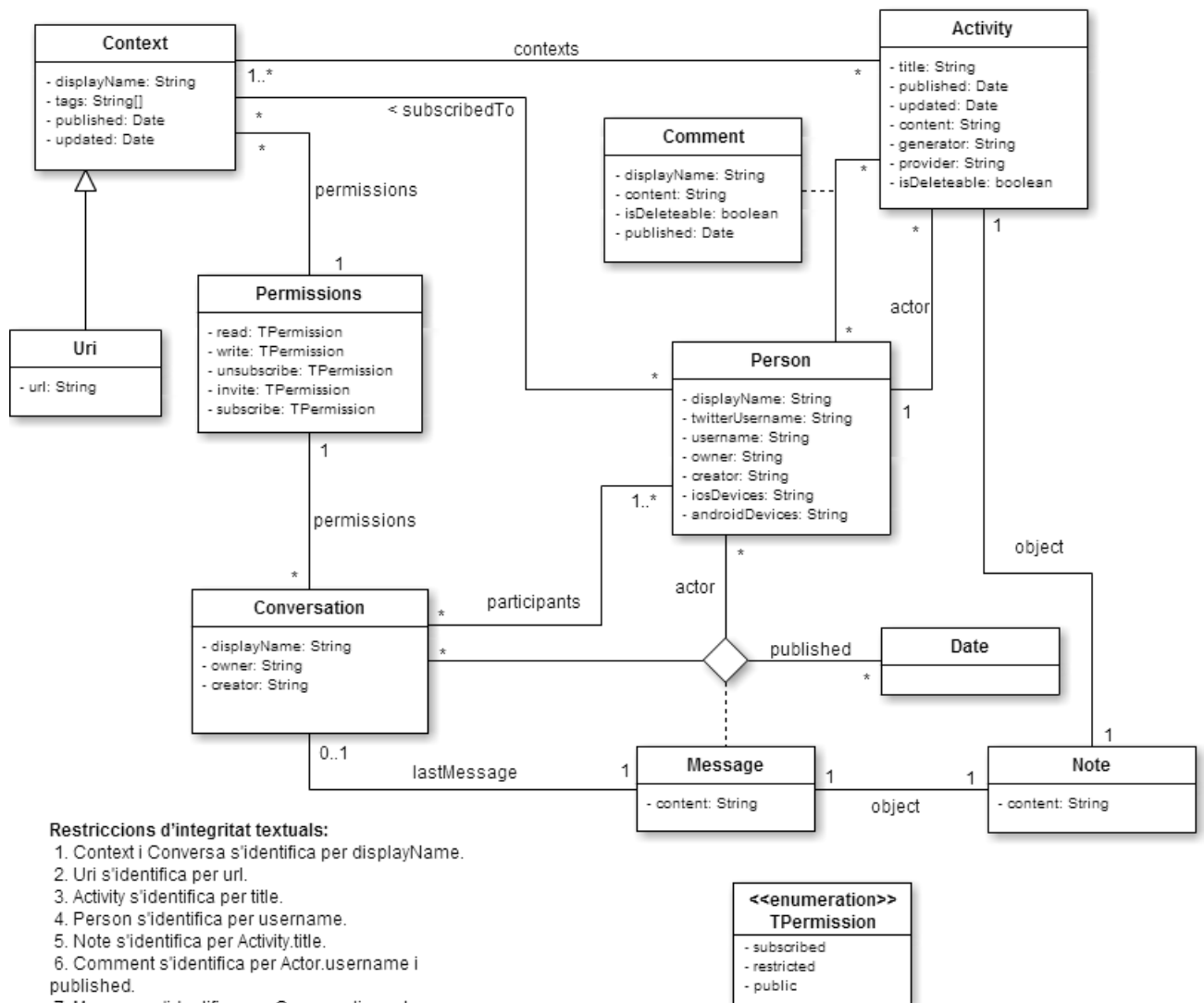


Figura 7: Model conceptual de l'aplicació.

### 4.1.3 Model de comportament

Totes les operacions del sistema es poden agrupar segons operacions d'obtenció d'informació, de publicació de contingut, d'esborrat o de modificació.

Com que totes les operacions són molt similars, només s'ha especificat el model de comportament per a una operació de cada agrupació.

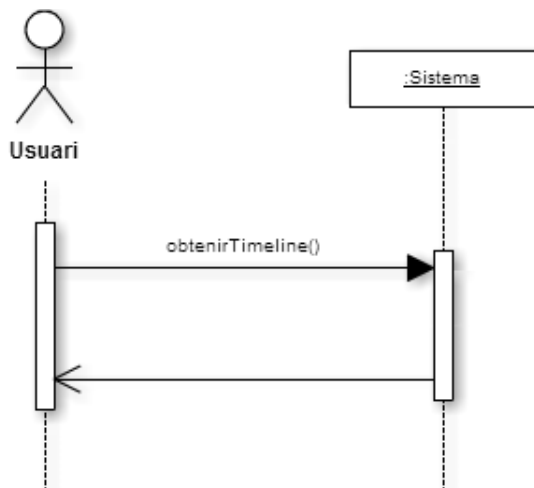


Figura 8: Diagrama de seqüència d'obtenció d'informació.

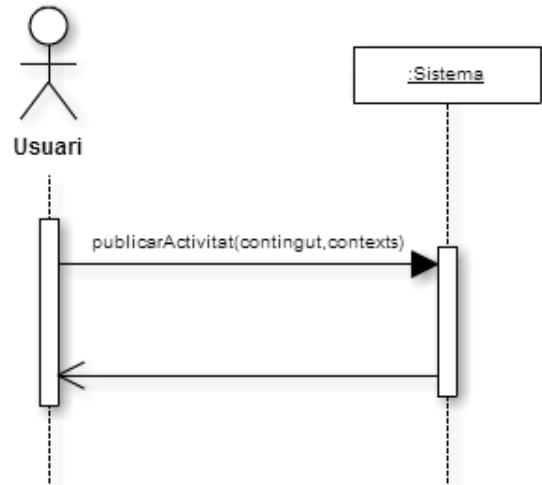


Figura 9: Diagrama de seqüència de publicar contingut.

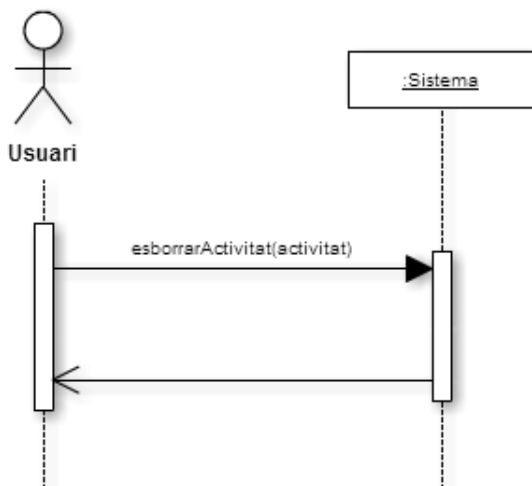


Figura 10: Diagrama de seqüència d'esborrat.

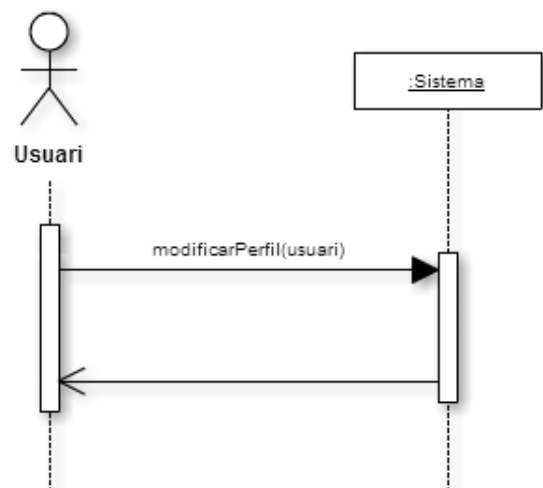


Figura 11: Diagrama de seqüència de modificació.



## 4.2 Requisits no funcionals

A més dels requisits funcionals de l'apartat anterior, el sistema ha de complir una serie de requisits no funcionals. Aquests requisits estan definits seguint una plantilla simplificada que inclou la següent informació:

- Identificador
- Nom del requeriment
- Descripció del requeriment

A continuació es pot veure primer un exemple del format, i després tots els requisits no funcionals del sistema.

|                    |  |
|--------------------|--|
| <b>ReqNF. #:</b>   | Identificador                              |
| <b>Nom:</b>        |  |
|                    | Nom del requeriment                        |
| <b>Descripció:</b> |  |
|                    | Descripció i justificació del requeriment. |

|                    |   |
|--------------------|---|
| <b>ReqNF. #:</b>   | 1   |
| <b>Nom:</b>        |   |
|                    | Usable  |
| <b>Descripció:</b> |   |
|                    | El sistema ha de ser senzill d'utilitzar i intuïtiu per als usuaris als que va dirigit. |

|                    |   |
|--------------------|---|
| <b>ReqNF. #:</b>   | 2   |
| <b>Nom:</b>        |   |
|                    | Multi-idioma  |
| <b>Descripció:</b> |   |
|                    | El sistema ha d'estar disponible en Català, Espanyol i Anglès. I ha de ser fàcil de traduir a altres idiomes. |

|                    |   |
|--------------------|---|
| <b>ReqNF. #:</b>   | 3   |
| <b>Nom:</b>        |   |
|                    | Robust  |
| <b>Descripció:</b> |   |
|                    | El sistema no ha de permetre l'entrada per part de l'usuari de dades incorrectes. Per tal de mantenir la integritat del sistema, també ha de garantir que no es produeixin duplicitats. |

|                    |  |
|--------------------|--|
| <b>ReqNF. #:</b>   | 4  |
| <b>Nom:</b>        |  |
|                    | Complir la <i>App Review Guidelines</i>  |
| <b>Descripció:</b> |  |
|                    | El producte s'ha d'adaptar a la guia de revisió d'aplicacions d' <i>Apple</i> ( <i>App Review Guidelines</i> [4]). |

|                    |   |
|--------------------|---|
| <b>ReqNF. #:</b>   | 5   |
| <b>Nom:</b>        |   |
|                    | Canviable   |
| <b>Descripció:</b> |   |
|                    | El sistema ha de permetre realitzar canvis al sistema de manera que aquests no afectin a les funcionalitats ja existents sense requerir molt temps per a realitzar-los. |

|                    |  |
|--------------------|--|
| <b>ReqNF. #:</b>   | 6  |
| <b>Nom:</b>        |  |
|                    | Disseny UPC  |
| <b>Descripció:</b> |  |
|                    | L'aplicació ha de tenir un disseny d'acord amb el disseny institucional de la UPC, i per tant ha de seguir les guies d'estil de la UPC[8]. |

|                    |   |
|--------------------|---|
| <b>ReqNF. #:</b>   | 7   |
| <b>Nom:</b>        |   |
|                    | Extensible  |
| <b>Descripció:</b> |   |
|                    | El sistema ha de ser fàcilment extensible. Ha de ser possible afegir fàcilment noves funcionalitats així com millorar les funcionalitats existents. |

|                    |   |
|--------------------|---|
| <b>ReqNF. #:</b>   | 8   |
| <b>Nom:</b>        |   |
|                    | Compatible última versió iOS  |
| <b>Descripció:</b> |   |
|                    | El sistema ha de ser com a mínim compatible amb la última versió del sistema operatiu iOS. Intentant mantenir el màxim de compatibilitat amb versions anteriors del sistema operatiu. |



## 5 Disseny arquitectònic

### 5.1 Patrons de disseny

Les aplicacions iOS estan basades en *Cocoa*. *Cocoa* és una interfície de programació d'aplicacions natives orientada a objectes per a sistemes operatius OS X. En aquesta secció estan explicats els patrons de disseny per a aplicacions basades en *Cocoa*.

#### 5.1.1 Model Vista Controlador

El principal patró de disseny utilitzat per les aplicacions basades en *Cocoa* és el patró Model Vista Controlador (MVC). És un patró de disseny de software que separa la interfície d'usuari, la lògica de control i les dades de l'aplicació, com es pot veure a la figura 12. Aquest patró incrementa la re-utilització i la flexibilitat del projecte.

Esta format per:

- Un Model
- Varies Vistes
- Varis Controladors

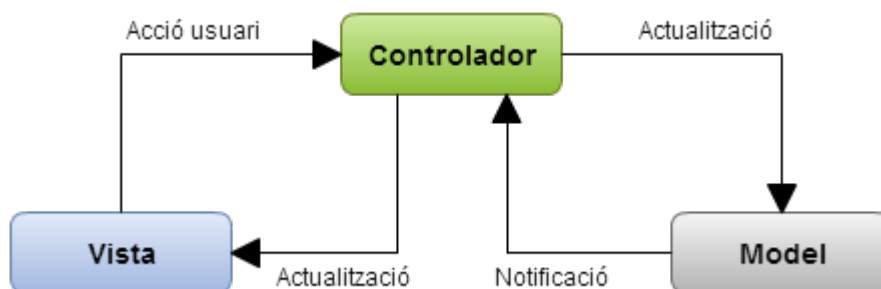


Figura 12: Flux de control del patró MVC.

El Model representa la informació amb la que el sistema treballa. Les Vistes presenten el model amb el que interactua l'usuari, també conegut com a interfície. El Controladors gestionen els esdeveniments que es capten a les vistes, normalment generats per accions que l'usuari realitza i que impliquen canvis en el model i a la mateixa interfície.

A la figura 12 podem veure el flux de control del patró Model Vista Controlador.

- L'usuari realitza una acció en una vista.

- El controlador tracta l'acció d'entrada, que pot implicar un canvi en l'estat del model.
- El controlador si ho necessita genera una nova vista i s'espera a una altre nova acció de l'usuari.

El patró MVC és un patró que marca tres rols diferenciats i la comunicació que hi ha d'haver entre ells.

### 5.1.2 Notificacions

El patró Notificacions permet la comunicació entre objectes sense implicar que aquests estiguin acoblats. Un objecte pot enviar informació a els altres objectes sense informació específica d'aquests objectes. Els objectes que estan interessats en rebre alguna informació es registren utilitzant una instància de de la classe `NSNotificationCenter`<sup>12</sup>. S'anomenen observadors els objectes registrats, és per això que sovint aquest patró de disseny també és anomenat patró Observador.

Quan s'envia una notificació al centre de notificacions, aquest distribueix la notificació a tots els observadors que s'han registrat a aquell tipus de notificació.

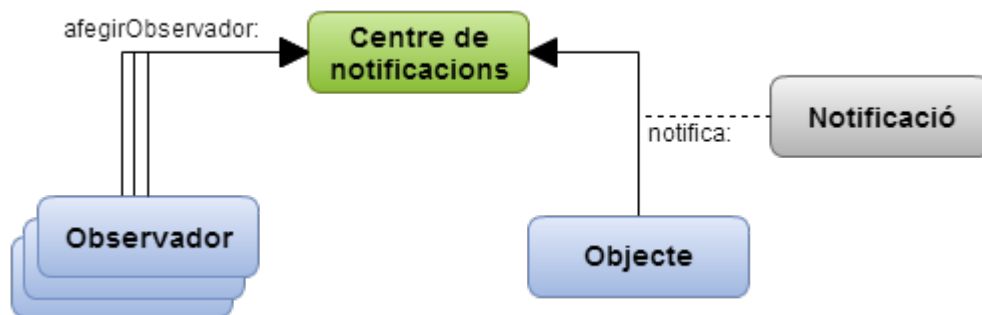


Figura 13: Comunicació entre el centre de notificacions i els observadors.

L'objecte que envia un missatge cap al centre de notificacions no necessita conèixer quins observadors existeixen ni quants n'hi ha. De manera similar, els observadors no necessiten conèixer l'origen de les notificacions rebudes. A la figura 13 es mostra la comunicació entre el centre de notificacions i els observadors, així com la comunicació amb l'objecte que envia la notificació.

<sup>12</sup>Centre de notificacions de la llibreria d'*Apple*

### 5.1.3 Categoria

El patró de disseny Categoria ofereix la possibilitat d'afegir mètodes a classes ja existents. Ofereix una alternativa per a fer subclasse en algunes situacions. Aquest patró està suportat directament pel llenguatge Objective-C. Ha estat utilitzat a la implementació d'altres patrons de disseny basats en *Cocoa*.

Pot ser utilitzat per substituir la implementació d'alguns mètodes en classes existents. S'utilitza freqüentment per a organitzar la implementació d'algunes classes. En aquest projecte s'ha utilitzat per afegir comportaments a les classes de la llibreria *Cocoa*, per exemple per a afegir a la classe *NSDate* la hora "en format humà" (fa 5 minuts, fa 2 dies).

### 5.1.4 Delegat

El delegat és un patró de disseny simple i molt potent. L'objecte que s'ha delegat manté una referència cap a l'altre objecte (el delegat) i li pot enviar missatges. El missatge informen d'algun fet succeït, donant l'oportunitat al delegat de reaccionar davant d'aquests fets. El delegat pot respondre al missatge actualitzant el seu estat o l'estat d'altres objectes de l'aplicació. El principal valor de la delegació és que permet personalitzar fàcilment el comportament de diversos objectes amb un objecte central.

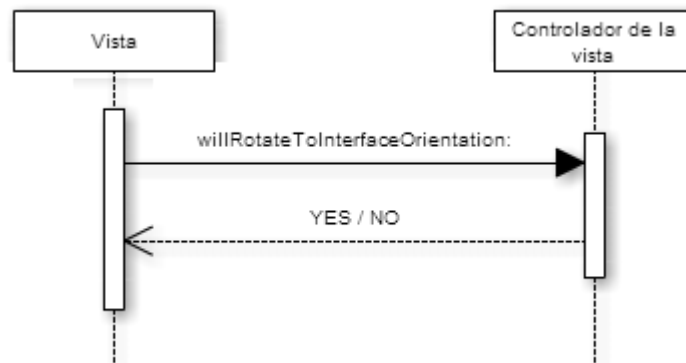


Figura 14: Comunicació vista-controlador mitjançant delegats.

Un exemple d'objecte delegat és un controlador de vista d'una aplicació d'*iOS*. Totes les vistes tenen declarat un protocol, on uns dels mètodes delegats és "willRotateToInterfaceOrientation:". Quan el dispositiu va a rotar, la vista afectada per la rotació li envia al seu delegat "willRotateToInterfaceOrientation" per a que el controlador de vista respongui si ha de girar o no. Es pot veure un esquema en la figura 14.

### 5.1.5 Singleton

El patró de disseny *Singleton* és utilitzat per a tenir una única instància d'una classe i aquesta ha de ser fàcilment accessible per altres objectes de l'aplicació. Habitualment, s'implementa amb un mètode que crea una nova instància de la classe en el cas que aquesta no existeix. En el cas en que si que existeixi, retorna una referència al objecte.

Aquest patró de disseny en aquest projecte s'ha decidit aplicar als controladors de suport (el controlador del MAX i de RestKit). D'aquesta manera tot objecte que necessiti accedir al MAX podrà referència a la classe *singleton* sense necessitar configurar-ho cada cop.



## 5.2 Protocols

En aquest projecte no es disposa de capa de dades ja que tota la informació s'obté sempre mitjançant el servidor MAX. Per aquest motiu els protocols, estàndards i especificacions utilitzats tenen una especial importància. A continuació es defineixen els més importants que s'han utilitzat a l'aplicació.

### 5.2.1 Activity Streams

L'*Activity Streams*[1] és una especificació d'un format obert per a protocols de flux d'activitat. Aquests s'utilitzen per a combinar activitats extretes d'aplicacions i serveis web socials, similars al mur d'activitat de *Facebook* o el *timeline* de *Twitter*.



Figura 15: Logo *Activity Streams*.

S'han realitzat implementacions d'aquesta especificació com a proveïdors i com a consumidors. *Facebook* i *MySpace* són un exemple de proveïdors i *Windows Live* un exemple de consumidor.

En aquest projecte el servidor MAX desenvolupat per UPCnet és un proveïdor d'*Activity Streams*. I l'aplicació iOS client del MAX que s'ha desenvolupant en aquest treball un consumidor d'*Activity Streams*.

### 5.2.2 REST

Tècnica d'arquitectura de software per a sistemes hipermèdia distribuïts com la web. Tot hi que aquest terme també s'utilitza per a descriure qualsevol interfície web simple que utilitza *XML / JSON* i *HTTP*, sense les abstraccions addicionals de protocols basats en patrons d'intercanvi de missatges com el protocol *SOAP*.

El món web ha escalat correctament gracies a seguir un conjunt de dissenys fonamentals, segons REST.

- Protocol client-servidor sense estat: cada missatge *HTTP* conté tota la informació necessària per a resoldre la petició, no depen de peticions anteriors.
- Conjunt d'operacions que s'apliquen a tots els recursos d'informació: *HTTP* defineix les operacions *GET*, *POST*, *PUT* i *DELETE* que t'utilitzen per a obtenir, modificar, afegir i esborrar.

- Sintaxis universal per a identificar els recursos. En un sistema REST, es pot accedir a cada recurs unicament a partir de la seva *URI*.

Aquest terme es va originar a l'any 2000 en una tesis doctoral, l'autor és un dels principals autors de l'especificació del protocol *HTTP*, Roy Fielding.

### 5.2.3 STOMP

*Simple Text Oriented Message Protocol* o *Streaming Text Oriented Message Protocol*[20] anteriorment conegut com TTMP, és un protocol de missatgeria orientat a missatges de text. Té l'objectiu de facilitar la compatibilitat de la missatgeria entre els llenguatges i les plataformes.



Figura 16: Logo STOMP.

És un protocol independent del llenguatge de programació.

Per tant qualsevol sistema pot comunicar-se amb un altre sistema sense dependre d'estar implementats amb el mateix llenguatge.

En aquest projecte s'utilitza el protocol STOMP per a implementar les converses en temps real. El servidor MAX ho té implementat amb *websockets*.

## 5.3 Capa de presentació

### 5.3.1 Disseny extern

El disseny extern d'aquest projecte ve fixat pel client que proporciona imatges del resultat desitjat realitzades per un dissenyador gràfic. A més proporciona tots els elements gràfics que s'han utilitzat per a realitzar la imatge.

A la figura 17 és pot comparar el pas del disseny al producte final. A l'esquerra de la figura es pot veure la imatge aportada pel client, i a la dreta una captura del resultat final.

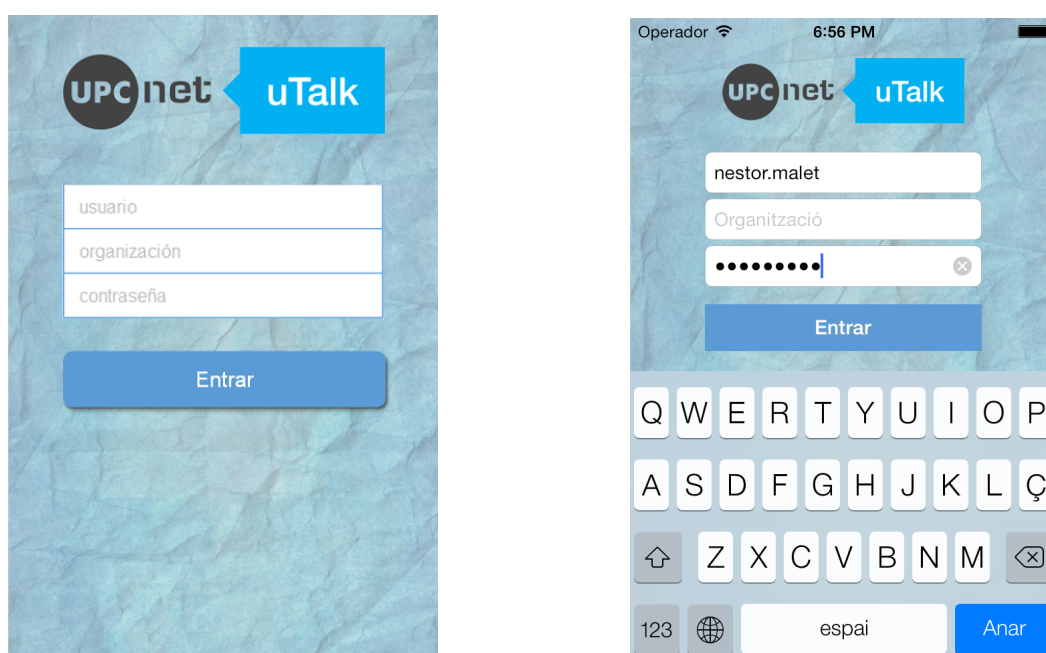


Figura 17: Comparació del disseny i el resultat de la vista

Excepte a la vista d'inici de sessió, tot el disseny extern del sistema segueix la mateixa estructura. A la figura 18 es pot veure aquesta estructura amb els elements més importants marcats amb colors.

L'element principal que organitza tot el sistema, són les pestanyes. A la figura aquest element està marcat amb color blau. Dintre de cada pestanya, la navegació es realitza mitjançant la barra de navegació, que a la figura està marcada amb color verd. I per acabar, a l'element de color vermell es mostra el contingut de cada vista.

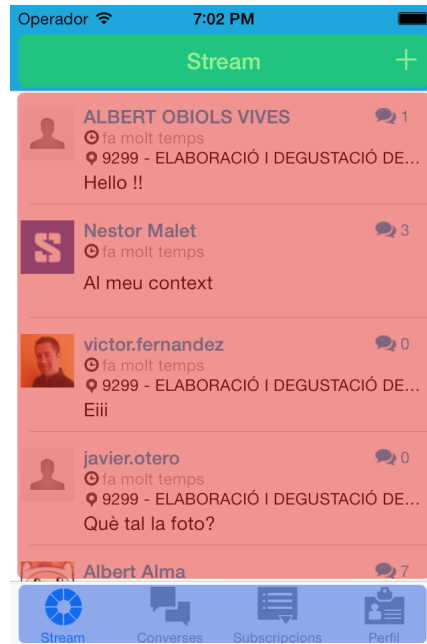


Figura 18: Estructura del disseny extern del sistema

### 5.3.2 Mapa de navegació

L'aplicació està dividida en pestanyes, a continuació es poden veure els models de navegació de cada una de les pestanyes del sistema.

Un cop l'usuari ja ha iniciat sessió, el punt d'entrada inicial és el de *stream* de la pestanya del *timeline* (figura 19). Quan l'usuari no té la sessió iniciada, el punt d'entrada és el de *iniciarSessió* de la figura 22.

Al model de navegació de les subscripcions i del perfil (figures 21 i 22), hi ha accions que mostren vistes que estan definides al model de navegació del *timeline* (figura 19).

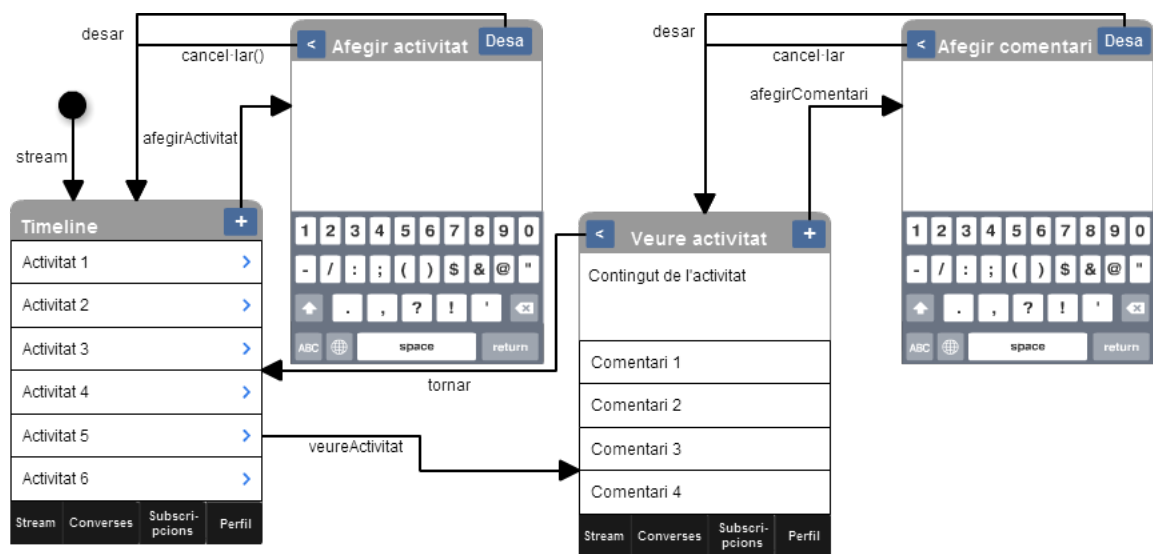


Figura 19: Mapa de navegació del *timeline*.

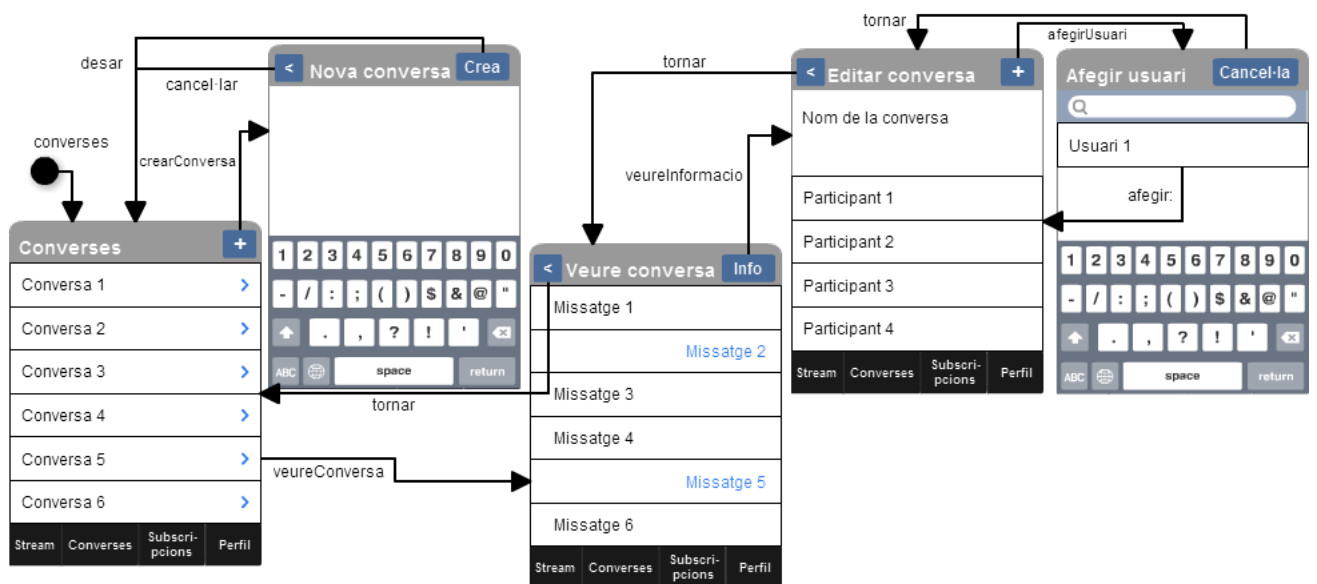


Figura 20: Mapa de navegació de les converses.

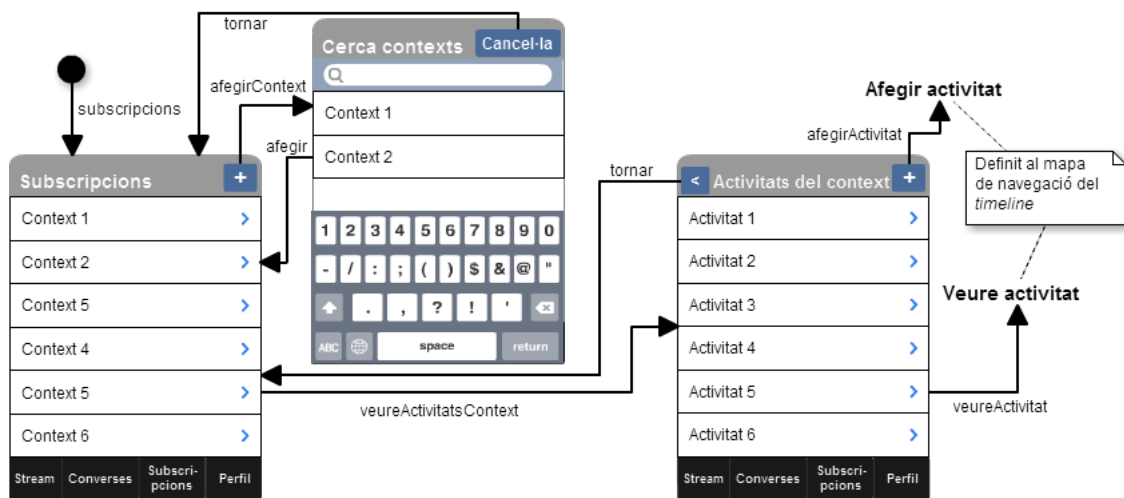


Figura 21: Mapa de navegació de les subscripcions.

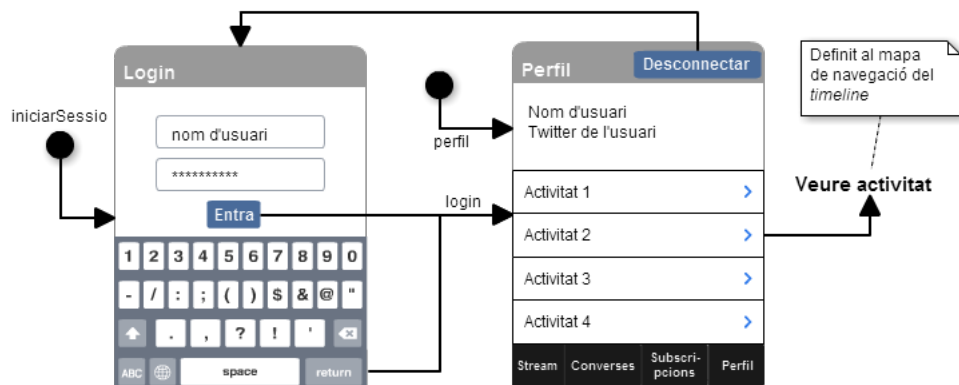


Figura 22: Mapa de navegació del perfil d'usuari.

## 5.4 Capa de domini

### 5.4.1 Model conceptual

Com a conseqüència d'aplicar la metodologia *scrum*, el disseny arquitectònic del model conceptual ha anat evolucionant durant el transcurs dels *sprints*.

A continuació es pot veure el model conceptual normalitzat resultant de l'aplicació.

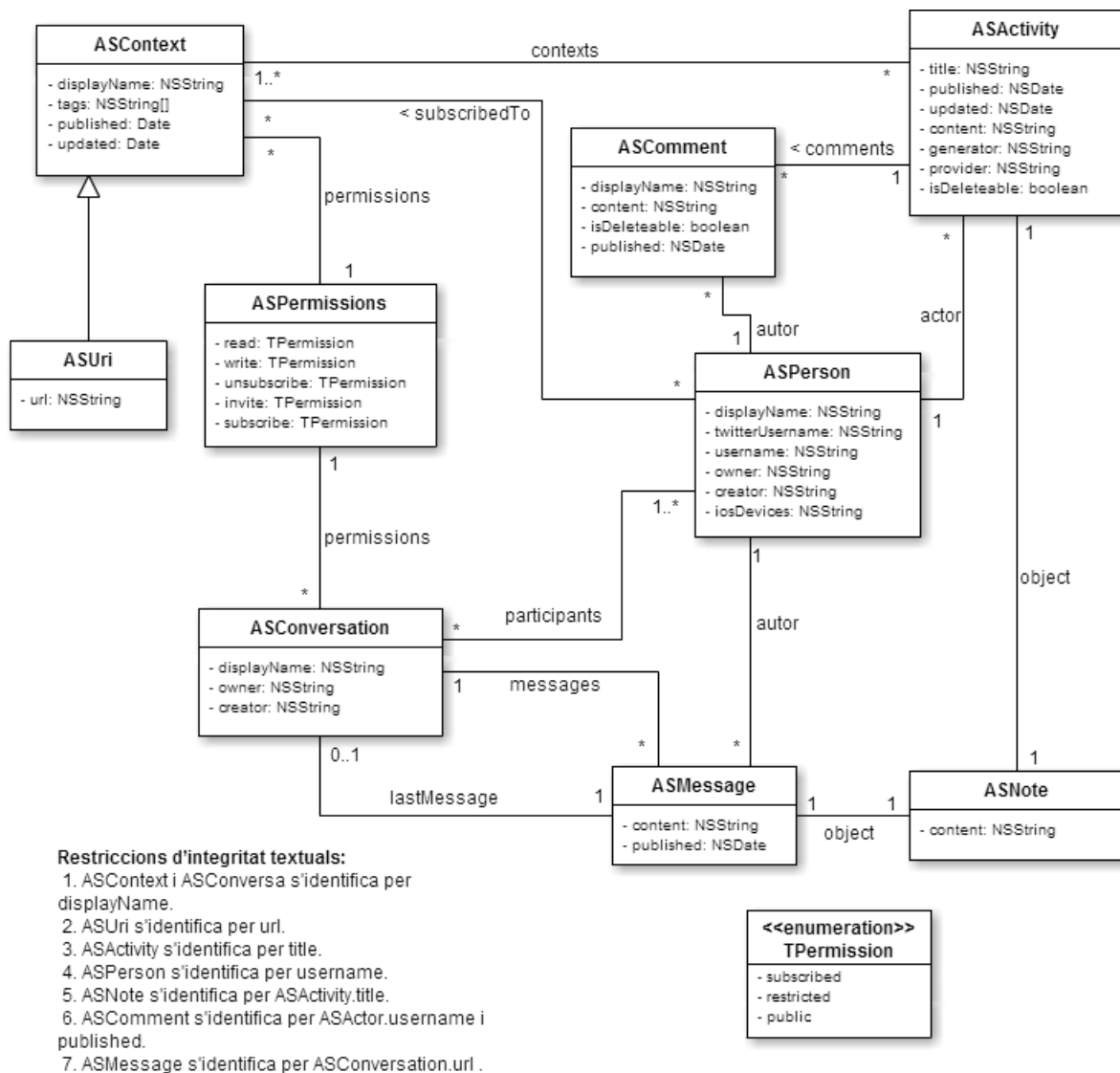


Figura 23: Model conceptual normalitzat.

### 5.4.2 Model de comportament

En aquesta memòria, s'ha inclòs un model de comportament per cada un dels quatre tipus d'operacions que té el sistema, seguint amb la tria que s'ha fet a la secció d'especificació del sistema.

El model conceptual segueix el patró de disseny *Domain Model*, ja que es delega a la classe responsable de cada acció la responsabilitat de realitzar-la.

La classe *UPCMaxController* és la responsable de fer les crides al servidor. Les consultes al servidor es realitzen amb crides asíncrones mitjançant l'us de delegats (el patró delegat està explicat a la secció 5.1.4), es delega al *UPCMaxController* la tasca de gestionar la resposta del servidor.

El *UPCMaxController* quan rep una resposta del servidor envia una notificació (el patró notificació està explicat a la secció 5.1.2) del tipus pertinent, per exemple *timeline* actualitzat amb èxit o error al carregar el *timeline*. I les vistes que s'han registrat com a observadores de la notificació, la reben i realitzen els canvis pertinents a la vista.



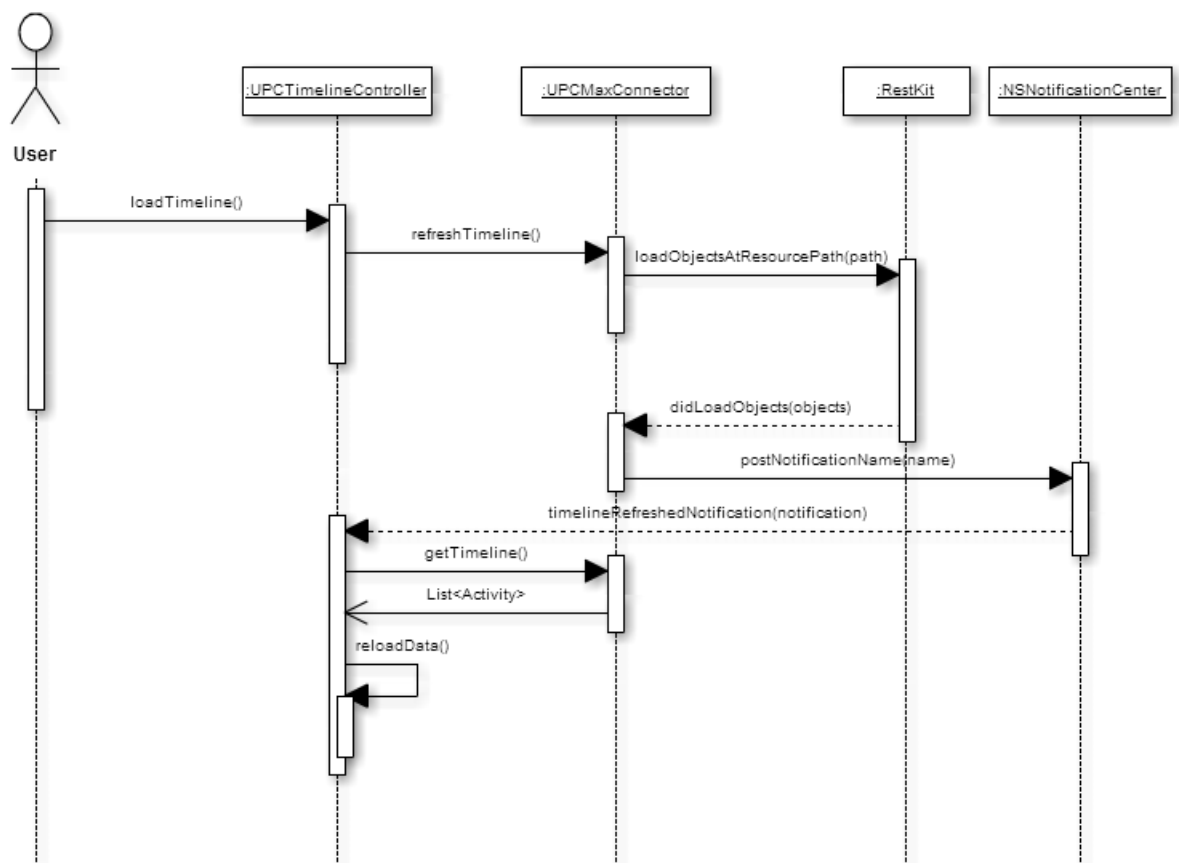


Figura 24: Diagrama de seqüència d'una operació d'obtenció d'informació.

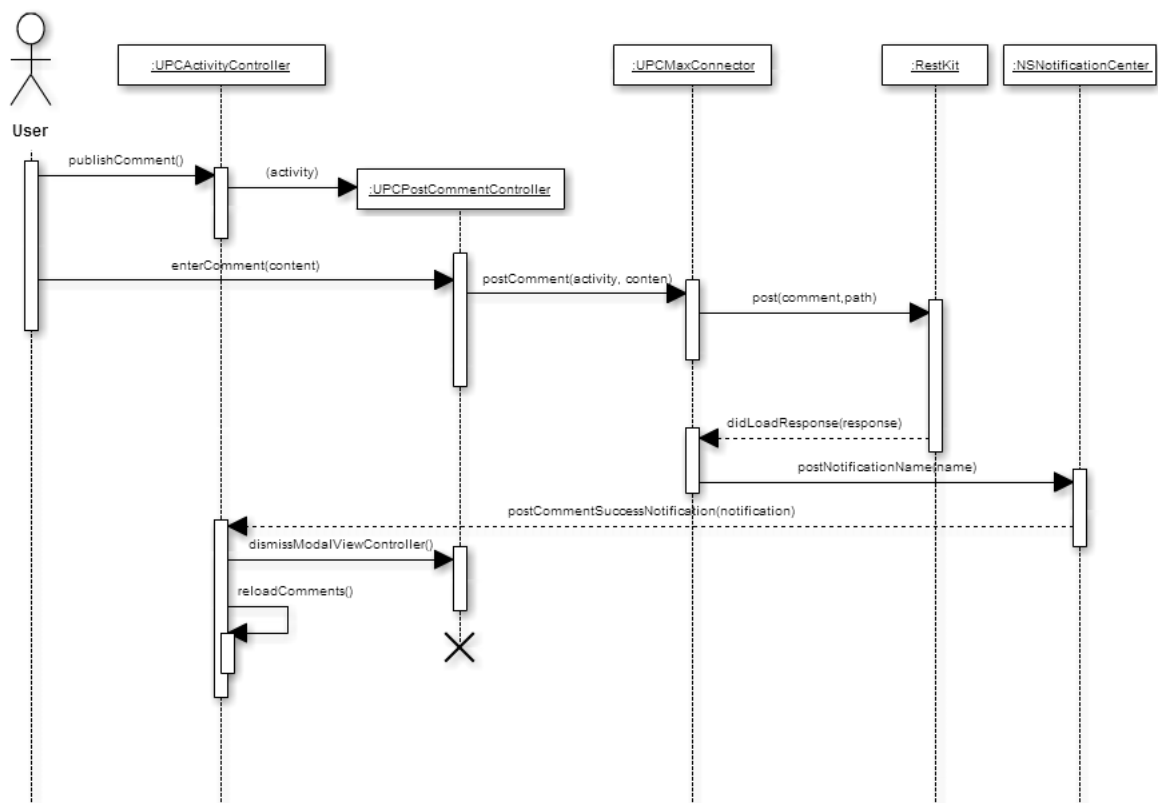


Figura 25: Diagrama de seqüència d'una operació de publicació d'informació.

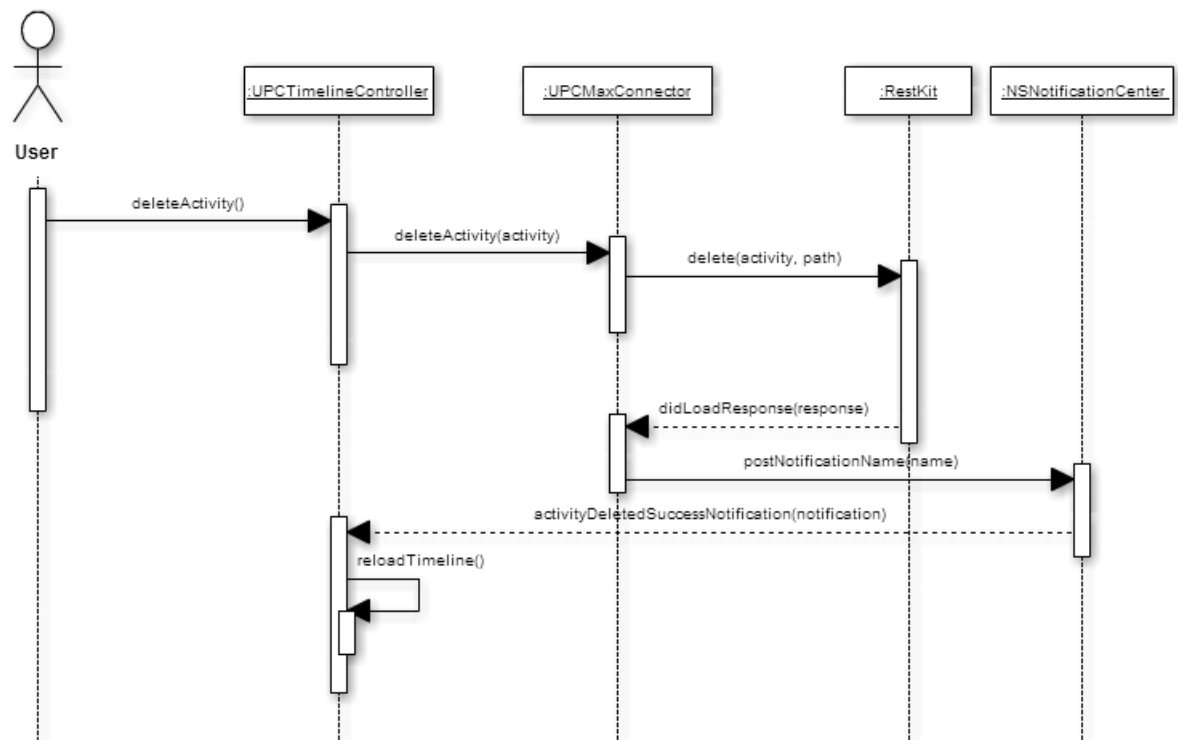


Figura 26: Diagrama de seqüència d'una operació d'esborrat.

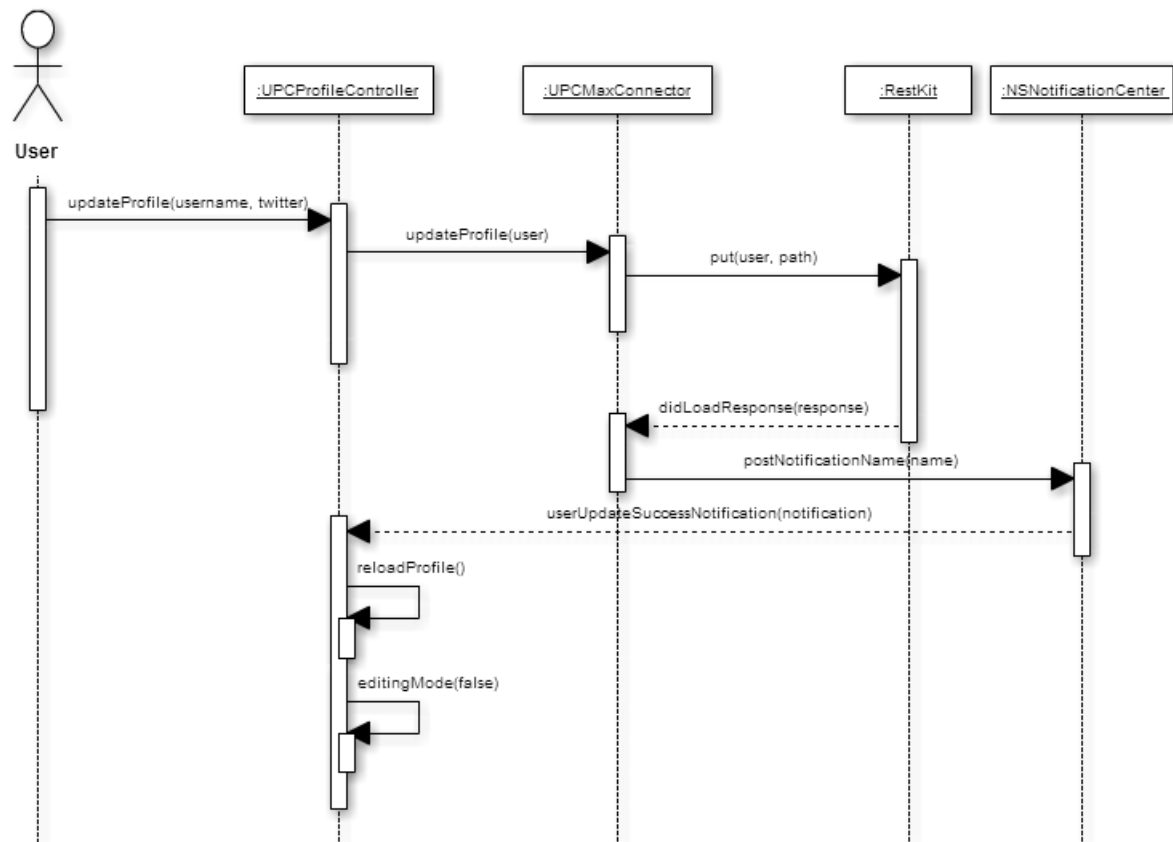


Figura 27: Diagrama de seqüència d'una operació de modificació de dades.

## 5.5 Capa de dades

Aquest projecte no disposa de capa de dades, ja que totes les dades les obté del servidor MAX. Aquestes dades no es guarden a l'aplicació i per tant cada cop es tanca i es torna a obrir l'aplicació, les dades es tornen a carregar del servidor.

La única informació que es guarda a l'aplicació són un conjunt de dades de l'usuari. Es persisteix el nom d'usuari i el *token*, que s'obtenen a l'hora d'iniciar sessió. Aquestes dades es guarden utilitzant *NSUserDefaults*<sup>13</sup>.

<sup>13</sup>Emmagatzematge que ofereix el sistema per a guardar parelles clau-valor.

## 5.6 Model de desplegament

El model de desplegament mostra els components del sistema a l'entorn de producció i com es relacionen entre ells.

El projecte d'aquest treball, forma part del SomUPC. Per aquest motiu el model de desplegament s'ha ampliat amb algunes components del model de desplegament del SomUPC per a complementar la visió de l'entorn de desplegament. A la figura 28 el dispositiu "Aplicació nativa", equival a l'aplicació client del MAX iOS (objecte d'aquest treball) o al client *Android* del MAX.

Els dispositius que estan directament relacionats amb aquest projecte són l'aplicació nativa iOS, el servidor MAX del que s'obté la informació social, el servidor OAuth amb el que es validen els usuaris i s'obté el *token* de l'usuari i el servidor SomUPC del que s'obtenen les imatges dels usuaris. A més el servidor MAX envia notificacions *push* mitjançant el servidor *APNS* d'*Apple*.

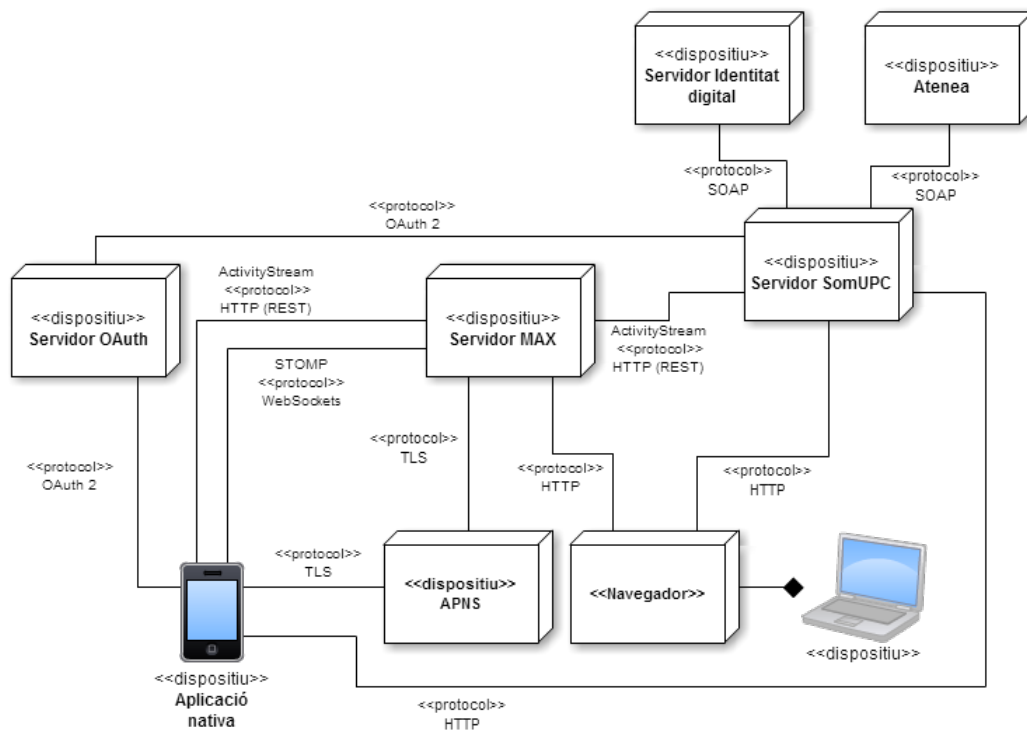


Figura 28: Model de desplegament.

## 5.7 Distribució de l'aplicació

Al tractar-se d'una aplicació iOS, el procés de distribució esta definit per *Apple*. A més en aquest projecte s'utilitza la plataforma *TestFlightApp*<sup>14</sup> per a realitzar proves amb usuaris reals. Els passos per a desplegar una versió de l'aplicació son els següents:

1. Instal·lar la versió al dispositiu a través del cable USB, i verificar el correcte funcionament.
2. Distribuir la versió mitjançant *TestFlightApp* per a que un grup d'usuari reduït (20-30 usuaris) verifiquin el correcte funcionament de la versió.
3. Enviar la versió al sistema de distribució d'*Apple*.
4. Un cop *Apple* revisa i accepta la versió, es distribueix mitjançant l'*App Store* d'*Apple*.

Per tant hi ha tres canals per a instal·lar l'aplicació a un dispositiu, a través de l'*App Store* d'*Apple*, a través de *TestFlightApp* o mitjançant un cable USB. A la figura 29 es poden veure especificats els tres canals de distribució.

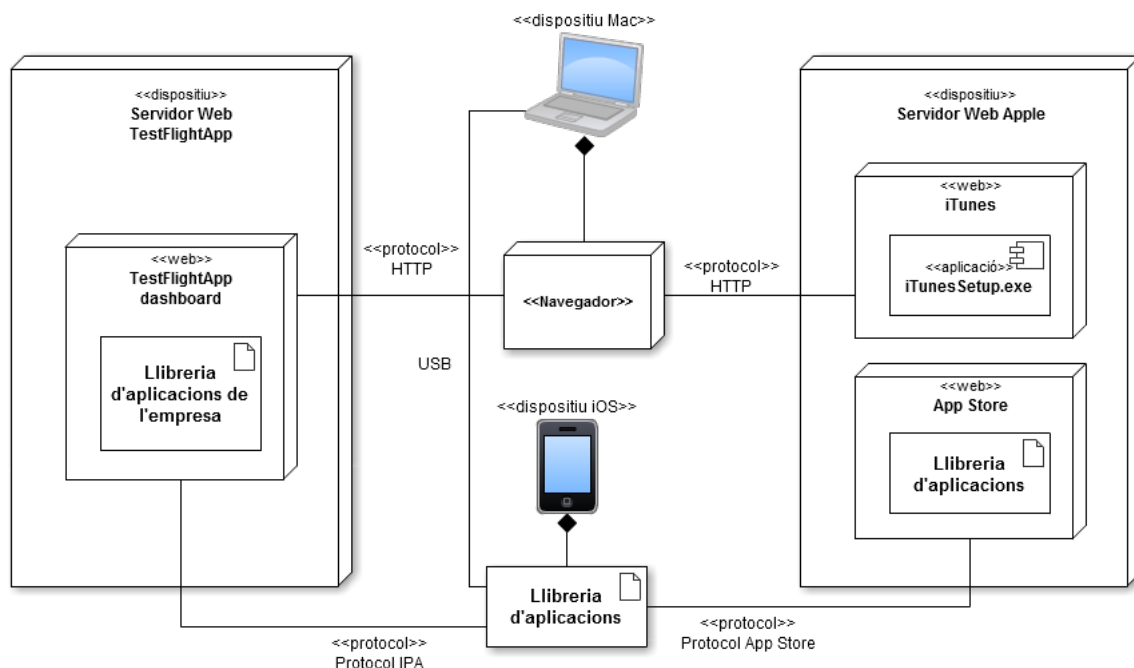


Figura 29: Distribució de l'aplicació.

<sup>14</sup>Sistema de distribució de versions *beta* de l'aplicació. <https://testflightapp.com/>

## 6 Desenvolupament

En aquesta secció s'explica el desenvolupament del projecte, l'anàlisi i implementació de cada una de les històries d'usuari definides a l'especificació del projecte.

Per a facilitar la comprensió, s'ha realitzat una agrupació d'històries d'usuari en etapes. Aquesta agrupació s'ha realitzat en funció de la temàtica de la història. Això no implica que s'hagi desenvolupat el projecte en l'ordre que apareixen les històries a continuació.

A la taula 7 es poden veure les històries d'usuari que s'han desenvolupat a cada *sprint*. A l'annex, a la secció 10.2, estan les històries d'usuari per *sprint* amb més detall.

| 1 | 2           | 3  | 4       | 5       | 6 | 7             | 8        | 9  | 10      | 11       | 12                   | 13 | 14 | 15 | 16 |
|---|-------------|----|---------|---------|---|---------------|----------|----|---------|----------|----------------------|----|----|----|----|
| 1 | 2<br>3<br>4 | 15 | 8<br>12 | 5<br>16 | 6 | 7<br>17<br>14 | 18<br>19 | 20 | 9<br>13 | 10<br>11 | 23<br>24<br>25<br>26 | 27 | 21 | 22 | -  |

Taula 7: Histories d'usuari realitzades per *sprint*.

### 6.1 Estructura de l'aplicació

L'estructura de l'aplicació és la primera etapa, i prerequisite de la resta d'etapes. L'objectiu és el d'obtenir un esquelet base d'aplicació iOS per a poder implementar les funcionalitats que es demanen. Aquesta base ha de tenir les següents funcionalitats:

- El sistema ha de demanar a l'usuari que introdueixi les seves credencials.
- Les dades que ha introduït l'usuari s'han de validar amb el servei d'autenticació.
- Si les dades són correctes s'ha d'obtenir el *token* de l'usuari, si no són correctes s'ha d'informar a l'usuari del problema.
- El sistema ha d'establir la connexió amb el MAX amb el *token* de l'usuari.
- El sistema ha de poder obtenir informació d'algun dels serveis del MAX, com per exemple el *timeline*.

Aquesta etapa inclou les històries d'usuari 1 i 2 (pàgina 39).

### 6.1.1 Implementació

A l'iniciar aquesta etapa es partia de zero ja que era l'inici del projecte. Abans de començar a crear el projecte es va fer un petit estudi de la situació, per poder decidir quina estructura era la millor i com fer la connexió al MAX.

L'estructura de la interfície està fixada pel client que va demana que l'aplicació estigui estructurada en pestanyes, i dintre de les pestanyes s'ha de navegar utilitzant la barra de navegació.

Sobre l'estructura interna de l'aplicació es va decidir seguir els patrons habituals de disseny que proposa *Apple*[5]. Aquests patrons en essència són:

- *Model-View-Controller* (Model-Vista-Controlador, MVC) <sup>15</sup>: Aquest patró de disseny s'utilitza per a estructurar tota l'aplicació.
- *Delegation* (Delegació) <sup>16</sup>: Facilita la transmissió d'informació i dades d'un objecte a un altre.
- *Target-action* <sup>17</sup>: Patró de disseny que relaciona les interaccions de l'usuari (botons i controls) amb el codi que la aplicació ha d'executar.
- *Block objects* <sup>18</sup>: Utilitza blocs de codi per implementar *callbacks* i codi asíncron.

Un dels punts més importants va ser prendre una decisió sobre com connectar el sistema amb el MAX. Per fer-ho es van considerar tres possibilitats:

- **Fer les crides des de l'aplicació**: consisteix en realitzar totes les consultes i modificacions des de l'aplicació i d'aquesta manera fer tot el sistema molt eficient. Aquesta solució té l'inconvenient de presentar un alt nivell d'acoblament entre l'aplicació i el MAX, i per tant, el codi resultant seria molt poc canviable.
- **Implementar una llibreria**: aquesta possibilitat implica desenvolupar una llibreria externa a l'aplicació que permeti obtenir la informació del MAX mitjançant una API en Objective-C. Aquesta solució ofereix un bon nivell de canviabilitat ja que els canvis en el MAX només comportarien canvis a la llibreria. Aquesta opció implica un gran

---

<sup>15</sup><https://developer.apple.com/library/ios/documentation/General/Conceptual/DevPedia-CocoaCore/MVC.html>

<sup>16</sup><https://developer.apple.com/library/ios/documentation/General/Conceptual/DevPedia-CocoaCore/Delegation.html>

<sup>17</sup><https://developer.apple.com/library/ios/documentation/General/Conceptual/DevPedia-CocoaApp/TargetAction.html>

<sup>18</sup><https://developer.apple.com/library/ios/documentation/General/Conceptual/DevPedia-CocoaCore/Block.html>



esforç de desenvolupament.

- **Utilitzar una llibreria que faciliti la connexió:** en aquest cas s'hauria de fer una cerca de possibles llibreries i posteriorment triar la més adient per a aquest cas. L'objectiu és delegar a la llibreria desenvolupada per un tercer la responsabilitat d'establir i mantenir la connexió amb el MAX, així com fer totes les consultes i modificacions seguint la filosofia REST. El principal inconvenient d'aquesta opció és el de limitar el desenvolupament a les funcionalitats que ofereix la llibreria.

Es va decidir descartar la primera opció ja que el producte resultant seria difícil de mantenir. Per decidir entre la segona i la tercera opció es va fer una petita cerca de llibreries de tercers.

Com a resultat de la cerca es van obtenir quatre llibreries que cumplien amb els requisits:

- ASIHTTPRequest <sup>19</sup>: aquesta llibreria facilita l'establiment de connexions amb el servidor per a fer consultes oferint una API de més alt nivell que les consultes natives d'Objective-C. El problema principal d'aquesta llibreria és que no té un suport continuat ni s'adapta a les necessitats del projecte.
- AFNetworking <sup>20</sup>: llibreria similar a ASIHTTPRequest, facilita l'establiment de connexions amb una API de més alt nivell que la nativa. De la mateixa manera que amb l'anterior llibreria, aquesta no és la més adient per al projecte ja que només facilita l'accés al servidor, però no facilita la implementació dels principis REST.
- RestKit <sup>21</sup>: aquesta llibreria proporciona una interfície completa per a que l'aplicació es comuniqui amb un servidor REST de forma semi-transparent. És a dir que l'aplicació ha de configurar tots els mapeigs dels serveis del servidor i posteriorment l'aplicació només ha de fer les crides a RestKit, que les transformarà en crides REST.
- Spaghetti <sup>22</sup>: llibreria similar a RestKit però per servidors que no utilitzen el protocol REST, per tant aquesta llibreria és més flexible que la anterior.

Després d'estudiar i analitzar totes les possibilitats es va decidir fer una prova de concepte amb la llibreria RestKit[19]. Perquè era la que més s'ajustava a les necessitats del projecte i podia aportar més valor.

La prova de concepte va consistir en crear el projecte de l'aplicació. Es va partir d'una

---

<sup>19</sup><http://allseeing-i.com/ASIHTTPRequest/>

<sup>20</sup><https://github.com/AFNetworking/AFNetworking>

<sup>21</sup><https://github.com/RestKit/RestKit>

<sup>22</sup><https://github.com/noodlewerk/Spaghetti>

plantilla d'aplicació amb pestanyes que ofereix *Apple*. Es va afegir la llibreria RestKit i es va provar de configurar amb un *token* del MAX posat manualment. El primer servei que es va intentar utilitzar va ser el del *timeline*. Es va configurar el mapeig del servei i el mapeig del model de dades (seguint l'estàndard *Activity Streams*, explicat a la secció 5.2.1).

Una vegada obtingudes amb èxit les dades del MAX del *timeline* mitjançant la prova de concepte, es va decidir utilitzar RestKit per al projecte.

A partir d'aquí es va continuar amb el desenvolupament afegint la pantalla d'inici de sessió. Per a validar l'usuari no es va utilitzar RestKit, si no que es va fer una consulta directament amb el servei de validació de la UPC ja que aquest no funciona amb REST si no que segueix el protocol d'*OAuth2*[17]. A la figura 30 es pot veure a quins àmbits s'utilitza RestKit i a quins no s'utilitza.

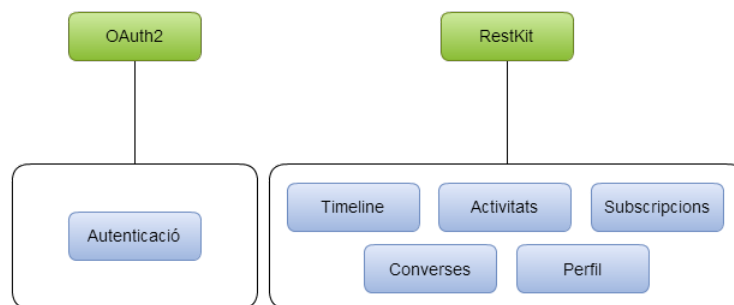


Figura 30: Limit d'ús de la llibreria RestKit.

Un cop l'usuari estava validat s'obtenia el *token* i ja es podia configurar RestKit amb el nom d'usuari i el *token* de l'usuari que havia iniciat sessió.

## 6.2 *Timeline* i veure activitat

Aquesta etapa té com a objectiu implementar les històries d'usuari relacionades amb el *timeline* i les activitats. Com la resta de les etapes, aquesta només es pot realitzar si s'ha finalitzat la primera etapa (Estructura de l'aplicació). En aquesta etapa s'han d'implementar les següents funcionalitats:

- El sistema ha d'obtenir i mostrar les activitats del *timeline* de l'usuari.
- Si l'usuari selecciona una activitat al *timeline*, el sistema ha de mostrar el contingut d'aquesta. Ha de poder veure el text, la data de publicació, l'autor i els comentaris de l'activitat.
- L'usuari ha de poder publicar un nou comentari a una activitat.
- L'usuari ha de poder publicar una nova activitat, indicant el text i el context al que es publica.
- L'usuari ha de poder esborrar una activitat i un comentari.
- L'usuari ha de poder publicar una imatge. El sistema ha de mostrar una pre-visualització a sota del text de l'activitat.
- L'aplicació ha de ser capaç d'obrir fitxers des de qualsevol altra aplicació. Quan l'usuari ho realitzi, se li ha de mostrar la vista de publicació d'una activitat, indicant que està adjuntant un fitxer.

Aquesta etapa inclou les històries d'usuari 3, 4, 5, 6, 22, 23 i 27 (pàgines 40 i 46).

### 6.2.1 Implementació

Per completar la primera etapa es va obtenir la informació del *timeline* de l'usuari. Però aquesta informació no es mostrava amb la interfície, només es mostrava per consola. En aquesta etapa aquesta informació s'havia de mostrar seguint les directrius que havia establert el client.

A la pestanya del *timeline* es va afegir una taula amb les cel·les personalitzades. Es va decidir utilitzar la taula nativa d'iOS (*UITableView*) per a mostrar la informació de les activitats ja que d'aquesta manera es podien aprofitar totes les funcionalitats d'aquest element (per exemple la possibilitat d'esborrar un element lliscant amb el dit cap a un costat). En aquesta cel·la es mostra el nom de l'autor de l'activitat juntament amb la seva imatge, la data de publicació i un tros del text de l'activitat.

Un cop es va disposar de la vista del *timeline* completada, es va continuar afegint una nova vista per a mostrar la informació d'una activitat. Aquesta vista es mostra quan l'usuari clica

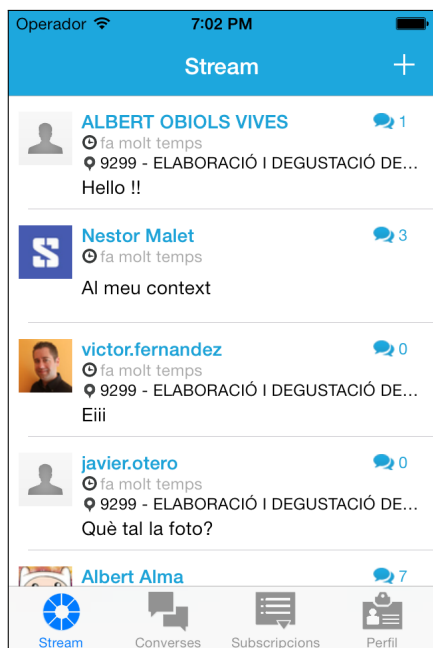


Figura 31: Resultat de la vista del *timeline*.

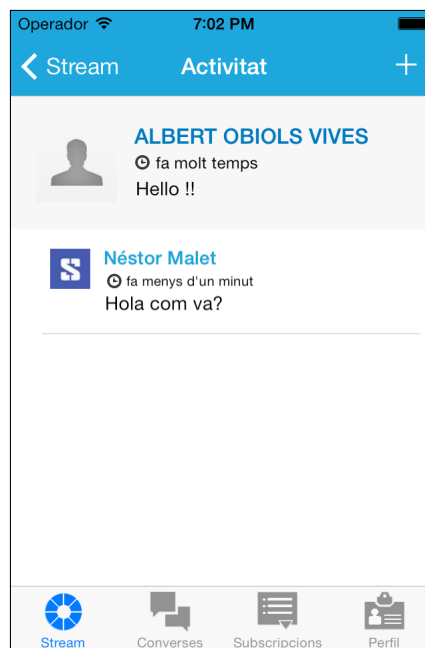


Figura 32: Resultat de la vista de veure activitat.

a una cel·la al *timeline*, aleshores el sistema carrega la informació de l'activitat del servidor i la mostra a l'usuari.

En aquesta nova vista, la vista de l'activitat, es mostra el nom de l'autor de l'activitat, la data de quan es va publicar i el text complert de l'activitat. Al costat d'aquesta informació es mostra la imatge de l'autor. En cas que l'activitat disposi de comentaris, aquests es mostren a sota en format de llista.

Per a posar els comentaris també es va optar per utilitzar una taula nativa d'iOS pels mateixos motius que al *timeline*. La funcionalitat d'afegir comentaris es va implementar amb una nova finestra que es mostra quan l'usuari clica al botó "+" de la barra de navegació.

Aquesta vista conté un camp de text i dos botons a la barra de navegació: "Desar" i "Cancel·lar". Si l'usuari després d'introduir un text el desa, es fa la petició al servidor per afegir el comentari. Un cop el servidor ha creat el comentari l'aplicació tanca la vista d'afegir un comentari i refresca els comentaris de la vista veure activitat.

Les funcionalitats esborrar activitat i esborrar comentari es van afegir utilitzant la funcionalitat de les taules natives d'iOS. Detectant l'acció de l'usuari de "lliscar" cap a la dreta una cel·la de la taula i mostrant un botó d'esborrar. Un cop l'usuari clica el botó, el sistema demana confirmació a l'usuari, i si l'usuari accepta, el sistema realitza la petició al servidor per a realitzar l'esborrat.

Per a implementar la confirmació que demana el sistema a l'usuari, es va utilitzar un diàleg

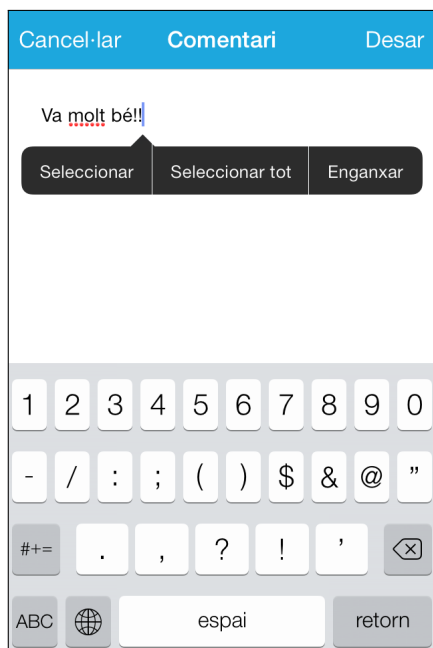


Figura 33: Resultat de la vista de publicar comentaris.

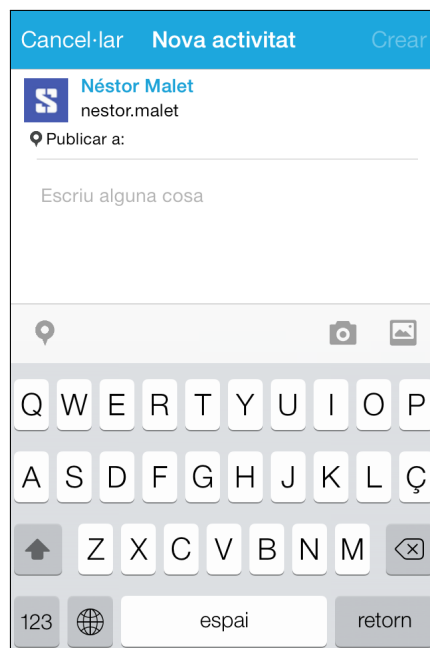


Figura 34: Resultat de la vista de publicar activitat.

natiu d'iOS (*UIAlertView*) amb dos botons, un per acceptar i l'altre per a cancel·lar.

Per a desenvolupar la funcionalitat d'afegir noves activitats, es va partir de la d'afegir comentari. Es va afegir a sobre del camp de text la informació de l'usuari i als contexts als que es publica.

El client va demanar que per triar els contexts als que publicar, es mostrés una nova vista amb la llista de contexts als que l'usuari té permís per a publicar. En aquesta vista l'usuari havia de poder fer una selecció múltiple de contexts i tornar a la vista de nova activitat.

Un cop l'usuari ha introduït el contingut de l'activitat i ha seleccionat els contexts als que publicar, el sistema fa les crides necessàries per a publicar les activitats al servidor. Per a obrir aquesta vista l'usuari ha de clicar al botó "+" de la barra de navegació del *timeline*.

Inicialment les activitats només havien de contenir text, però gracies a utilitzar la metodologia *scrum*, el client durant el transcurs del projecte va necessitar la funcionalitat d'afegir imatges/fitxers a les activitats, i per tant es va afegir com a funcionalitat requerida. Per afegir aquesta funcionalitat es van reajustar algunes funcionalitats, simplificant-les. No va ser necessari treure cap funcionalitat.

Aquest nou requisit implicava dues tasques, l'usuari havia de poder afegir una imatge des de l'aplicació (de la galeria d'imatges o de la càmera) i l'usuari havia de poder afegir fitxers des d'una altra aplicació del dispositiu.

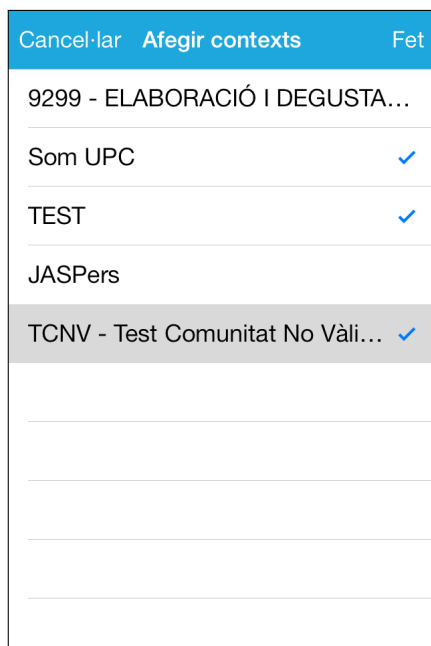


Figura 35: Resultat de la vista d'afegir contexts.

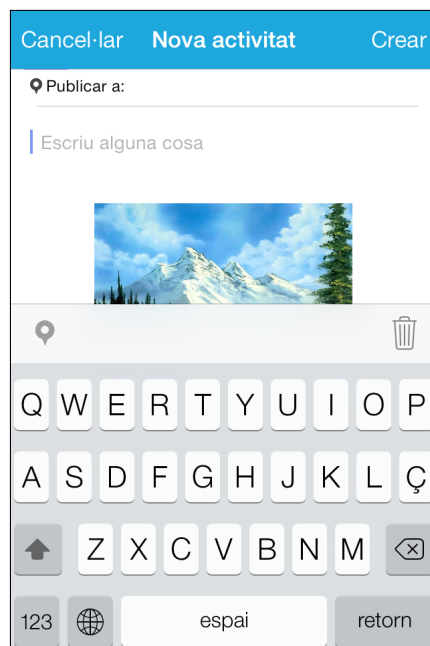


Figura 36: Resultat de la vista de publicar una activitat amb una imatge.

La primera tasca va implicar afegir dos botons a la interfície (un per la galeria i un per la càmera), que al clicar obri el diàleg natiu corresponent. Un cop l'usuari ha triat la imatge, aquesta s'afegeixi a sota del text de l'activitat en format miniatura. A més s'han de canviar els dos botons per un d'una paperera per a poder esborrar la imatge.

Per a la segona tasca, es va modificar el fitxer de configuració de l'aplicació. Es va declarar que l'aplicació pot obrir tot tipus de fitxers. A més, es va preparar l'aplicació per a que al obrir un fitxer amb l'aplicació, se li mostri a l'usuari la vista de nova activitat. A sota del camp de text de la vista, es mostra una capsa amb el nom del fitxer que s'està adjuntant.

En tots dos casos, quan l'usuari crea la nova activitat, es fan les crides corresponents al MAX per a enviar el fitxer.

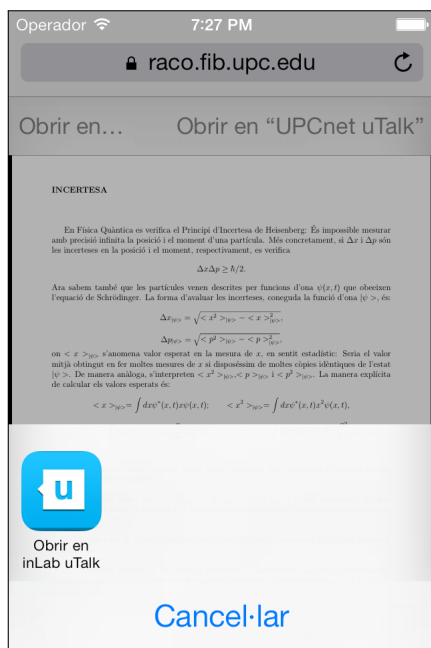


Figura 37: Adjuntar un fitxer des d'una altra aplicació.

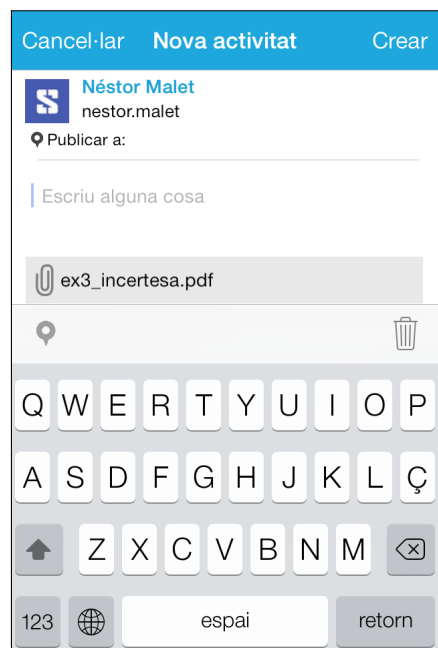


Figura 38: Resultat de la vista de publicar una activitat amb un fitxer.

## 6.3 Gestió de subscripcions

L'objectiu d'aquesta etapa és el de dotar a l'aplicació la funcionalitat de gestionar les subscripcions de l'usuari. És a dir, oferir a l'usuari la possibilitat de veure les seves subscripcions, esborrar-ne una i afegir-ne de noves.

Per a considerar l'etapa com a finalitzada, l'aplicació ha de tenir les següents funcionalitats:

- El sistema ha d'obtenir i mostrar les subscripcions de l'usuari.
- Si l'usuari selecciona una subscripció, el sistema ha de mostrar les activitats del context seleccionat.
- L'usuari ha de poder buscar contextos als que es pugui subscriure.
- Si l'usuari ha buscat un context i el selecciona s'ha de subscriure.
- L'usuari ha de poder esborrar una subscripció.

Aquesta etapa inclou les històries d'usuari 7, 8, 9, 10 i 24 (pàgines 41 i 47).

En aquesta etapa es poden aprofitar algunes vistes definides a “*Timeline* i veure activitat”. Com que les etapes no impliquen un ordre, en cas que el client prioritzi aquestes funcionalitats per davant de les de l'altra etapa es realitzaran en aquesta etapa.

### 6.3.1 Implementació

El primer que es va fer va ser afegir una nova pestanya a l'aplicació i en aquesta pestanya afegir una taula amb el seu controlador. Aquesta taula obtenia les dades des les subscripcions de l'usuari i les agrupava per tipologia del context.

Per a distingir les diferents tipologies dels contextos, aquests a l'atribut *tags* contenen una etiqueta pre-definides. A la taula 8 es poden veure les diferents etiquetes i els seus significats. Si un context no disposa de cap pre-definida, aleshores es classifica com a “Altres”.

| <b>Etiqueta</b> | <b>Significat</b>         |
|-----------------|---------------------------|
| [ASSIG]         | Context d'una assignatura |
| [INST]          | Context institucional     |
| [COMMUNITY]     | Context d'una comunitat   |

Taula 8: Significat de les diferents etiquetes.



Després es va crear una nova vista similar a la del *timeline* però carregant només les activitats d'un context. Aquesta vista es mostra quan l'usuari selecciona un context de la llista de subscripcions.

Gràcies a la gran similitud entre aquesta vista i la del *timeline*, es va poder re-utilitzar una part de la feina feta a l'etapa "*Timeline i veure activitat*". Es va utilitzar la cel·la del *timeline* ja que a les dues situacions es mostrava una activitat i es necessitava mostrar la mateixa informació. Com que s'havia de tornar a mostrar la informació de l'activitat al clicar a la cel·la, també es va re-utilitzar la vista de veure una activitat.



Figura 39: Resultat de la vista de subscripcions.



Figura 40: Resultat de la vista d'activitats d'un context.

També s'havia d'oferir a l'usuari la possibilitat de publicar una activitat al context. Es va aprofitar la vista d'afegir una activitat fent una petita modificació per a poder obrir la vista amb un context ja seleccionat.

Per acabar aquesta etapa, s'havia de poder esborrar una activitat i una subscripció a les activitats d'un context i a la vista de les subscripcions respectivament. Per fer-ho es va seguir la forma en que s'ha definit anteriorment.

## 6.4 Converses

L'objectiu d'aquesta etapa és el d'afegir les converses privades entre els usuaris. S'han d'implementar les següents funcionalitats:

- El sistema ha d'obtenir i mostrar les converses de l'usuari.
- Si l'usuari selecciona una conversa, el sistema ha d'obtenir tots els missatges i mostrar-los a l'usuari en format de xat.
- L'usuari ha de poder publicar un nou missatge a una conversa.
- L'usuari ha de poder veure la informació d'una conversa (nom i participants) i modificar-ho si és el creador de la conversa.
- L'usuari ha de poder crear una nova conversa indicant els participants i el contingut del primer missatge. En cas que hi hagi més de dos participants, el sistema ha de demanar a l'usuari el nom del grup.
- L'usuari ha de poder sortir d'una conversa o esborrar-la si és el creador.

Aquesta etapa inclou les històries d'usuari 14, 15, 16, 17, 18, 19 i 25 (pàgines 43 i 47).

S'ha separat en una altra etapa les millores a les converses de temps real i notificacions *push*. Aquesta separació és deguda a que ajuntar-les amb aquesta etapa complicaria molt aquest apartat. Per aquest motiu s'ha considerat separar-ho en dues etapes. Això no implica que al projecte s'hagin d'implementar per separat.

### 6.4.1 Implementació

La vista de les converses, igual que la del *timeline* i les subscripcions mostra una llista de elements, en aquest cas converses, per tant el format ideal és el d'una taula.

Per a les converses es va haver d'afegir un nou tipus de cel·la amb la informació de la conversa (nom, text i data de l'últim missatge, i la fotografia de l'usuari/grup).

La tasca principal d'aquesta etapa va ser la de desenvolupar la vista de la conversa, amb l'estil de xat. Per a fer-ho es van valorar diverses opcions. Es va fer una petita cerca de projectes que haguessin necessitat implementar un xat/SMS, i tutorials que parlessin del tema. El resultat va ser que hi havien molts projectes, molts d'ells amb el codi obert.

El projecte que més s'ajustava a les necessitats d'aquest projecte va ser *AcaniChat*<sup>23</sup>. Però integrar els components d'*AcaniChat* amb els del projecte no era senzill, així que es va decidir seguir els passos que havien seguit els desenvolupadors d'aquesta llibreria, ja

---

<sup>23</sup><https://github.com/acani/AcaniChat>



Figura 41: Resultat de la vista de converses.



Figura 42: Resultat de la vista dels missatges d'una conversa.

que al *ReadMe* d'aquesta citaven les fonts.

Per aquest motiu es va seguir el tutorial *Tutorial on SMS style bubbles*[7]. Després de seguir-lo i obtenir les bafarades amb un text, es va modificar per ajustar les bafarades amb els requisits del client. Aquest volia que es mostres l'autor (si no era el mateix usuari) i la data de publicació a més del contingut del missatge.

Un cop ja es tenia la vista preparada només faltava preparar les crides per a carregar la informació del servidor.

Des de la vista d'una conversa clicant un botó de la barra de navegació, s'ha de mostrar una nova finestra amb les dades de la conversa. A més, en cas que es tracti d'una conversa de grup i l'usuari sigui el creador de la conversa, també ha de poder modificar el nom del grup i afegir/treure participants.

Per oferir aquesta funcionalitat, es va afegir una nova vista modal amb el nom de la conversa i una llista de participants. Com que a la llista hi havia possibilitats que es poguessin eliminar participants de la conversa es va utilitzar una taula nativa d'iOS. Es va crear una nova vista de cel·la per a poder mostrar la fotografia de l'usuari, el seu nom i una etiqueta indicant si és el propietari o no.

Si l'usuari és el creador de la conversa pot esborrar un usuari, lliscant la cel·la del participant cap a un costat. Aleshores la cel·la es desplaça cap a un costat i apareix un botó d'esborrar. Si l'usuari clica al botó se li mostra una confirmació i si aquest ho accepta s'es-

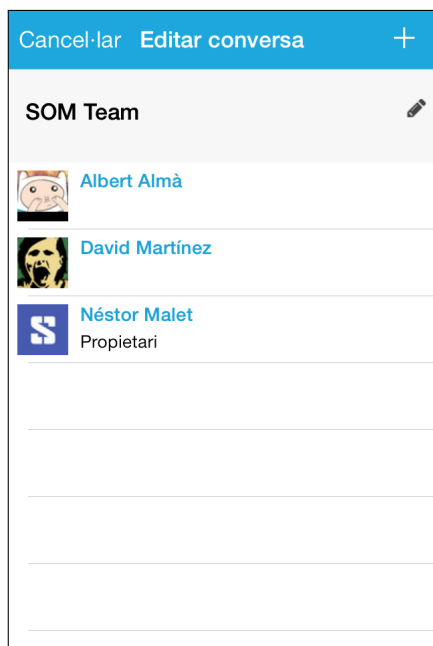


Figura 43: Resultat de la vista d'informació d'una conversa (com a propietari).

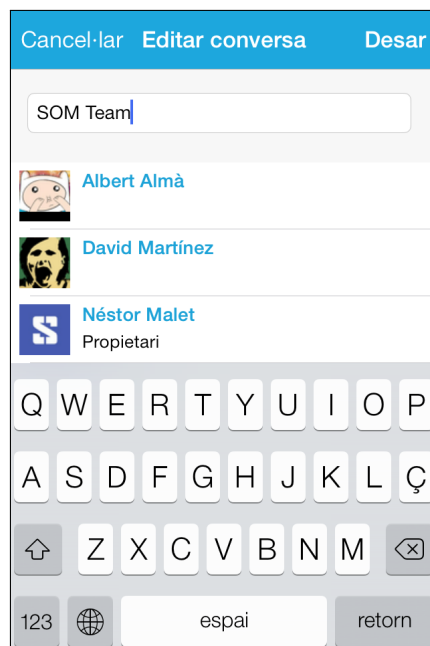


Figura 44: Resultat de la vista d'editar la informació d'una conversa.

borra el participant de la conversa.

En cas que l'usuari tingui permís per editar la conversa, al costat del nom del grup li apareix una icona d'un llapis. Si l'usuari clica a sobre del nom o del llapis, activa el mode d'edició.

En aquest mode l'etiqueta del nom de la conversa desapareix i es canvia per un camp de text. Però a més de poder editar el nom també pot afegir nous participants. Per fer-ho a la barra de navegació hi ha un botó amb un "+", quan l'usuari el clica se li mostra una vista modal amb un camp de cerca.

A la cerca, al mateix moment que l'usuari va escrivint el nom, el sistema realitza consultes i va mostrant el resultat a l'usuari. Un cop l'usuari selecciona un usuari de la llista de resultats, la vista de cerca es tanca. Aleshores es mostra un diàleg de confirmació per verificar que l'usuari realment vol afegir al membre.

La vista de cerca de persones es va desenvolupar com a component independent, per a poder ser re-utilitzat fàcilment (per exemple al crear una nova conversa). Per fer-ho, es va fer que el controlador de la vista retornés el *username* mitjançant una notificació (amb el *NSNotificationCenter*), i que la vista que havia instància el cercador estigués registrat com a observador de la notificació.

A la barra de navegació de la llista de converses, hi ha un botó "+" per a crear una nova conversa. En aquest cas, se li mostra una vista modal amb un camp de text per a introduir

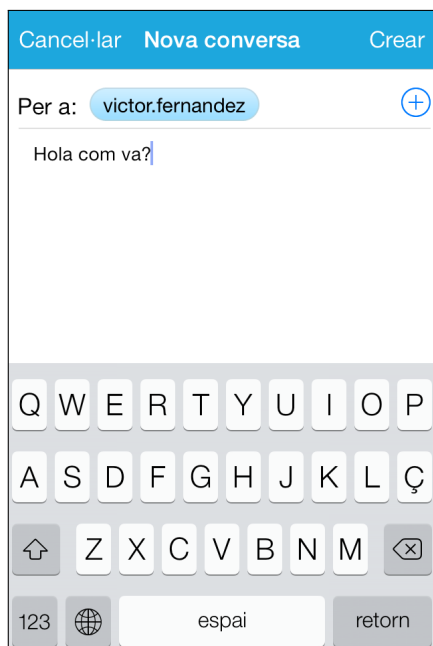


Figura 45: Resultat de la vista de nova conversa.

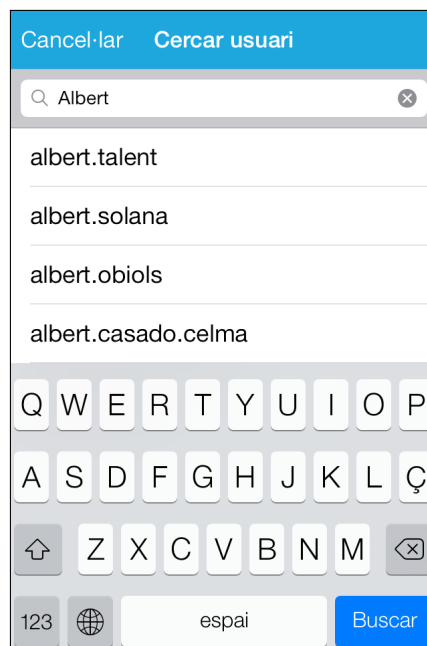


Figura 46: Resultat de la vista de cercar usuaris.

el contingut del primer missatge. A més a sobre d'aquest camp hi ha un camp per a afegir participants.

Els participants es poden afegir escrivint els noms d'usuari o clicant al botó "+". En aquest últim cas, el sistema mostra la vista de cercar usuaris (la mateixa que s'utilitza per afegir persones al editar una conversa).

A la llista de converses l'usuari pot esborrar o sortir d'una conversa, lliscant cap a un costat la cel·la. Un cop ho ha realitzat, el sistema mostra un botó d'esborrat. Abans d'esborrar la conversa, el sistema demana una confirmació a l'usuari.

## 6.5 Perfil

L'objectiu d'aquesta etapa és el d'oferir a l'usuari la possibilitat de veure el seu perfil i gestionar-lo. S'inclouen les següents funcionalitats:

- El sistema ha d'obtenir les dades de l'usuari i les seves activitats.
- Si l'usuari selecciona una activitat, el sistema ha de mostrar el contingut i els comentaris de l'activitat.
- L'usuari ha de poder modificar les seves dades personals (nom i nom d'usuari del *Twitter*).
- L'usuari ha de poder tancar la sessió.

Aquesta etapa inclou les històries d'usuari 11, 12, 13, 26 (pàgines 42 i 47).

En aquesta etapa es poden aprofitar algunes vistes definides a *Timeline* i veure activitat. Com que les etapes no impliquen un ordre, en cas que el client prioritzi aquestes funcionalitats per davant de les de l'altra etapa es realitzaran en aquesta etapa.

### 6.5.1 Implementació

La primera tasca d'aquesta etapa és la d'afegir una nova pestanya "Perfil" amb el seu controlador de vista.

En aquesta vista s'ha de mostrar la informació de l'usuari conjuntament amb la seva imatge. Per tant s'ha de carregar del MAX el perfil de l'usuari. Un cop està carregat s'ha de mostrar el nom, el nom d'usuari i el *twitter* de l'usuari.

A continuació s'han de mostrar les activitats publicades per l'usuari. Per fer-ho, seguint el patró definit prèviament, es va afegir una taula amb les cel·les personalitzades (re-utilitzant les del *timeline*). Es fa afegir la interacció de seleccionar una cel·la que obrirà la vista de veure activitat.

També es va afegir la interacció per esborrar les activitats, lliscant la cel·la cap a un costat. Quan l'usuari realitza aquesta acció, abans d'esborrar l'activitat, se li demana una confirmació per evitar esborrats accidentals.

El client va demanar que es pogués modificar el perfil de l'usuari. Per a fer-ho, es va establir que al costat del nom s'ha d'afegir una icona d'un llapis. Al clicar a sobre, la vista s'ha de transformar i a on abans es mostrava el nom ara s'ha de mostrar un camp de text amb el nom. De la mateixa manera s'ha de poder canviar el *twitter*.



Figura 47: Resultat de la vista del perfil.



Figura 48: Resultat de la vista del perfil en mode d'edició.

Quan el mode d'edició està activat, a la barra de navegació s'ha de mostrar un botó de cancel·lar i un de guardar. Si l'usuari clica al primer, s'han de descartar els canvis. Però si l'usuari clica al de guardar, s'ha de fer la modificació al servidor per a modificar el perfil de l'usuari amb els canvis introduïts per l'usuari.

Quan no està activat el mode d'edició, s'ha de mostrar un botó a la barra de navegació per a poder tancar la sessió de l'usuari. Un cop tancada la sessió, s'ha de mostrar la vista d'iniciar sessió.



Figura 49: Missatge de confirmació abans de tancar sessió.

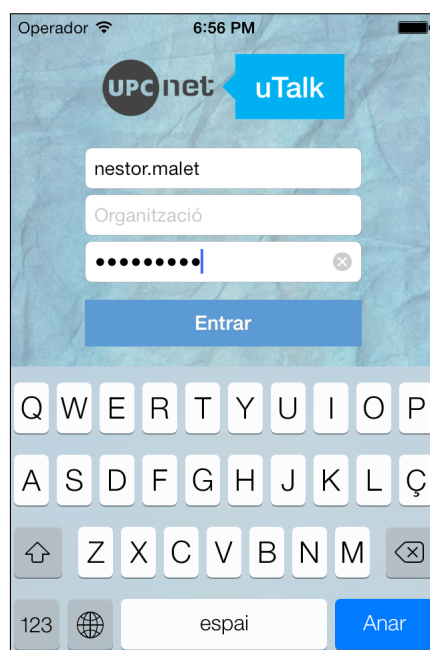


Figura 50: Resultat de la vista d'iniciar sessió.



## 6.6 Ampliació converses

Aquesta etapa realment és una continuació de l'etapa de "Converses", ja que s'afegeixen dues funcionalitats a les converses privades entre els usuaris. Les funcionalitats a desenvolupar són les següents:

- El sistema ha de notificar a l'usuari quan es publiqui un missatge a una conversa de l'usuari i aquest tingui l'aplicació tancada.
- Han d'arribar els missatges en temps real quan l'usuari tingui l'aplicació oberta.

Aquesta etapa inclou les històries d'usuari 20 i 21 (pàgina 45).

### 6.6.1 Implementació

Aquesta etapa està formada per dues grans tasques. Les dues funcionalitats tenen l'objectiu d'avisar a l'usuari quan un altre membre ha enviat un missatge a una conversa de l'usuari. La gran diferencia és la forma en que s'ha de notificar segons si té l'aplicació oberta o no.

Quan l'usuari té l'aplicació oberta, el missatge ha d'arribar en temps real, el dispositiu ha de vibrar i s'ha de mostrar a les vistes que ha arribat el missatge. Si l'usuari té oberta la conversa, ha de sortir el missatge a la llista de missatges. Si l'usuari no té oberta la conversa, s'ha de posar una icona a la pestanya de converses. Aquesta icona ha d'indicar el número de converses que tenen missatges sense llegir.

En canvi, quan l'usuari no té l'aplicació oberta, se li ha d'enviar una notificació *push* al seu dispositiu. Aquesta notificació ha de tenir la icona, el nom de l'aplicació, i l'inici del contingut del missatge que ha arribat.

Per a desenvolupar la primera tasca, la tasca que implica afegir el temps real, es va començar analitzant el sistema que el client ja tenia implementat i quines opcions existien per poder-ho adaptar a l'aplicació iOS del projecte.

El client UPCnet ja tenia implementat el sistema utilitzant *websockets*<sup>24</sup>. Normalment a una aplicació nativa que es comunica amb un servidor s'utilitzen *sockets*<sup>25</sup>, ja que el fet d'utilitzar *websockets* afegeix capes al procés i per tant redueix l'eficiència.

<sup>24</sup>Tecnologia que proporciona un canal de comunicació bi-direccional i *full-duplex* sobre un únic *socket* TCP

<sup>25</sup>Mecanisme per a l'entrega de paquets de dades provinents de la targeta de xarxa als processos. Queda definit per una parella de direccions IP local i remota, un protocol de transport i una parella de números de port local i remot.

Però com que el client ho va triar per a poder-ho utilitzar amb el *widget* web del MAX, i aquest no podia treballar amb *sockets*, o es mantenien les dues tecnologies o es treballava amb *websockets*. El client va desestimar la primera opció per evitar haver de mantenir les dues tecnologies alhora.

El sistema que el client tenia implantant utilitzava el protocol *STOMP* (explicat a la secció 5.2.3) per sobre dels *websockets*. Es va fer una cerca de llibreries que facilitessin l'ús d'aquesta combinació de protocol i tecnologia, però no se'n va trobar cap.

Es va trobar una llibreria que facilitava l'ús de *websockets* amb el llenguatge Objective-c, SocketRocket<sup>26</sup>. En concret aquesta llibreria oferia una API similar a l'API nativa de *sockets* d'Objective-c.

També es va trobar una llibreria client d'*STOMP* però que treballava amb *sockets*, obj-stomp<sup>27</sup>. Aquesta llibreria oferia una implementació completa del protocol *STOMP* per a Objective-c. Com s'ha dit prèviament, aquesta llibreria no ens servia ja que el nostre client tenia implementat el servidor amb *websockets*.

Com que les dues llibreries eren codi obert i es disposava el codi al *GitHub*, es va decidir re-implementar el codi de la segona (obj-stomp) per a que utilitzes *websockets*. Per a fer-ho es va realitzar un *fork* del projecte de *GitHub* d'obj-stomp<sup>28</sup>.

En aquest *fork*, es va canviar les crides de *sockets* per crides a la llibreria SocketRocket. Es van haver de fer alguns canvis a la constructora de la llibreria, ja que el *socket* necessita una IP i un port, en canvi el *websocket* necessita una direcció web. Un cop es va tenir la modificació de la llibreria, es va publicar al *GitHub* per a compartir-ho amb la comunitat de desenvolupadors.

A l'aplicació per a fer la gestió del temps real, es va decidir implementar una classe controladora del temps real (*UPCRealTimeManager*). Aquesta classe té la responsabilitat d'iniciar la connexió de temps real amb el servidor, i al rebre un missatge notificar-ho a la resta de l'aplicació. A més també és el responsable de tenir el control dels missatges que l'usuari no ha llegit.

Per realitzar aquesta notificació, es va decidir utilitzar el centre de notificacions. Es va crear un tipus de notificació de nou missatge, que a dintre portava com a contingut el nou missatge.

Totes les vistes que han de fer alguna acció al rebre un missatge, es registren com a

---

<sup>26</sup><https://github.com/square/SocketRocket>

<sup>27</sup><https://github.com/juretta/objc-stomp>

<sup>28</sup><https://github.com/nmaletm/objc-stomp>

observadores del tipus de notificació de nou missatge. Quan el centre de notificacions rep una notificació de nou missatge, aquest notifica a totes classes observadores.

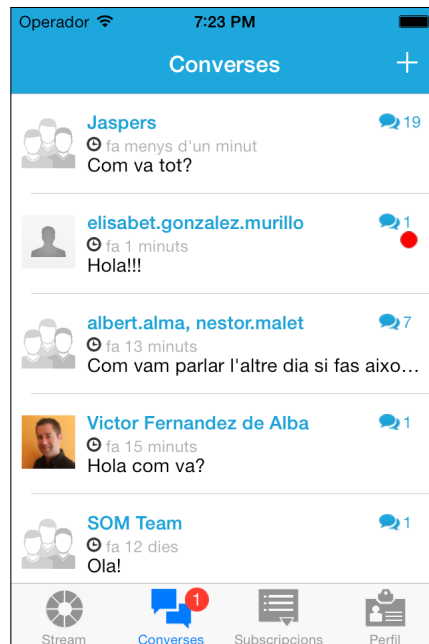


Figura 51: Llista de converses amb una conversa amb missatges sense llegir.

D'aquesta manera, cada vista definia el seu comportament en arribar un missatge. Per exemple, el controlador de la vista de veure conversa, al rebre un nou missatge, l'afegeix a la llista de missatges. A més aquest controlador cada cop que l'usuari veu una conversa, avisa al controlador de temps real que s'han vist els missatges d'aquella conversa.

Una altra vista que també depèn dels missatges nous, és la de llista de converses. Cada cop que arriba un nou missatge, aquesta marca la conversa com a conversa amb missatges sense llegir.

L'altra tasca d'aquesta etapa és la de rebre notificacions *push*. Aquesta tasca implica algunes accions per part de l'aplicació, però la majoria de tasques les ha de realitzar el servidor MAX.

En concret, l'aplicació l'únic que ha de realitzar és registrar i des-registrar l'identificador del dispositiu al MAX. Això és degut a que el MAX per poder enviar les notificacions necessita saber l'identificador del dispositiu al que ha d'enviar la notificació.

Aquest registre depèn del dispositiu i de l'usuari. Per tant, cada cop que s'inicia sessió s'ha d'afegir l'identificador al MAX per a l'usuari que ha iniciat sessió. De la mateixa manera, cada cop que es tanca la sessió, s'ha de treure l'identificador de l'usuari al MAX.

Però abans de poder afegir o treure l'identificador al MAX, aquest s'ha d'obtenir. Per fer-

ho es va preparar el *UPCAppDelegate* (delegat de totes les responsabilitats generals de l'aplicació), per a que sol·licités l'identificador i al rebre el guardés per a posteriors usos. Aquest identificador en aquest moment encara no estava vinculat a cap usuari, només estava vinculat al dispositiu.

Un cop ja es disposava de l'identificador, s'havia de preparar l'aplicació perquè en iniciar sessió i en tancar-la, registres/des-registres l'identificador al MAX al usuari.

També es va preparar el *UPCAppDelegate* perquè al rebre una notificació obrís la pestanya de converses.

A partir d'aquest moment l'aplicació ja podia rebre les notificacions *push*.

## 7 Línies de futur

Aquest projecte ha estat l'inici d'una aplicació que encara té molt camí per recórrer. És una de les peces del SomUPC i per tant alhora que aquest creixi, l'aplicació també ho farà. Aquest sistema s'implantarà properament a la Universitat Politècnica de Catalunya de mà de l'aplicació Android.

A més el client UPCnet implantarà imminentment el sistema a dues organitzacions, i continuarà buscant noves organitzacions on implantar el sistema.

En aquest projecte s'ha de continuar treballant afegint tots els canvis i millores que es facin al MAX. Així com arreglar tots els possibles problemes que es trobin a l'aplicació. Una altra gran evolució que aportarà molt valor a l'aplicació és afegir persistència.

Actualment si no hi ha connexió a Internet, l'usuari no pot veure cap dada a l'aplicació. Afegint persistència, es podria mostrar a l'usuari la informació que anteriorment s'hagués descarregat l'aplicació, en cas que aquest no tingués accés a la xarxa. També es podria mostrar informació antiga a l'usuari mentre que la nova es descarrega, especialment si l'usuari utilitza una connexió lenta a Internet.



## 8 Conclusions

Vaig començar aquest projecte sense saber que acabaria sent el meu Treball Final de Grau, durant la meua estada com a becari al inLab FIB. Aproximadament a mig projecte l'Albert em va oferir l'oportunitat de presentar-ho com a treball i després de fer tots els tràmits amb una setmana em vaig trobar immers a l'assignatura de GEP (Gestió de Projectes).

Abans de l'inici d'aquest projecte havia desenvolupat unes quantes aplicacions natives d'iOS, però mai m'havia trobat amb un projecte d'aquesta envergadura i abast. Estic molt content ja que m'ha ofert la possibilitat de veure créixer una aplicació des de zero i veure com organitzar l'estructura interna d'aquesta. Estic segur que es podria haver fet molt millor, però estic content del resultat obtingut.

Un altre aspecte que m'ha agradat molt d'aquest projecte, és que alhora que creixia l'aplicació iOS també ho feia l'aplicació Android. Gracies a que les dues evolucionaven a un ritme igual, els membres de l'equip, hem pogut comparar com afronten els mateixos problemes els dos sistemes. Ens hem adonat com, tot hi que puguin semblar iguals, els dos sistemes tenen plantejaments molt diferents i a vegades difícil fins i tot de comparar. En algunes ocasions hem pogut solucionar problemes d'un sistema aplicar patrons de l'altre sistema.

Al projecte SomUPC hem aplicat la metodologia *Scrum* i crec que ha estat una de les claus que han facilitat l'èxit del projecte. Al iniciar el projecte, teníem dues grans peces, el portal SomUPC i les aplicacions natives clients del MAX.

A l'inici es va prioritzar més el portal, però es va donar un punt en que les prioritats del client van canviar i es va prioritzar completament les aplicacions. Inclús durant un període de temps, es va paraitzar el portal fins acabar les aplicacions.

Aquest canvi, sota el meu punt de vista, era necessari, ja que el client per motius de negoci necessitava les aplicacions. Si no hagués estat per la flexibilitat de la metodologia potser no hauria estat possible. Si no s'hagués fet, el client segurament no s'hauria pogut queixar ja que el pacte inicial no contemplava el canvi. Però el fet de ser flexibles i adaptar-nos al client ha fet que aquest estigui content amb el projecte i li pugui donar continuïtat.

En aquest projecte hem treballat conjuntament l'equip SomUPC i UPCnet. Hem realitzat moltes reunions, l'experiència obtinguda en aquestes reunions és la major aportació que m'emporto d'aquest projecte. Estic segur que això m'aportarà molt al meu futur professional. Quan acabi la carrera i m'enfronti a la primera reunió de treball agrairé haver realitzat aquest projecte. Encara recordo la primera reunió que vam fer, i els nervis que vaig passar abans de començar.

Per acabar, m'agradaria agrair a tots els membres que han col·laborat en aquest projecte, tot l'equip SomUPC i l'equip d'UPCnet (negoci i equip tècnic).



## 9 Referencias

- [1] Activity Streams community. Estàndard Activity Streams community. <http://activitystrea.ms/>, 2013. [Online, accedit el 20-Setembre-2013].
- [2] Agencia Española de Protección de Datos. Cómo cumplir con la LOPD. [http://www.agpd.es/portalwebAGPD/jornadas/dia\\_proteccion\\_2011/responsable/index-ides-idphp.php](http://www.agpd.es/portalwebAGPD/jornadas/dia_proteccion_2011/responsable/index-ides-idphp.php), 2013. [Online, accedit el 15-Novembre-2013].
- [3] Agencia Tributaria. Regímenes para determinar el rendimiento de las actividades económicas. [http://www.agenciatributaria.es/AEAT.internet/Inicio\\_es\\_ES/\\_Segmentos\\_/Empresas\\_y\\_profesionales/Empresarios\\_individuales\\_y\\_profesionales/Rendimientos\\_de\\_actividades\\_economicas\\_en\\_el\\_IRPF/Regimenes\\_para\\_determinar\\_el\\_rendimiento\\_de\\_las\\_actividades\\_economicas/Estimacion\\_Directa\\_Simplificada.shtml](http://www.agenciatributaria.es/AEAT.internet/Inicio_es_ES/_Segmentos_/Empresas_y_profesionales/Empresarios_individuales_y_profesionales/Rendimientos_de_actividades_economicas_en_el_IRPF/Regimenes_para_determinar_el_rendimiento_de_las_actividades_economicas/Estimacion_Directa_Simplificada.shtml), 2012. [Online, accedit el 03-Juliol-2013].
- [4] Apple. App Review Guidelines. <https://developer.apple.com/appstore/guidelines.html>, 2013. [Online, accedit el 19-Juny-2013].
- [5] Apple. iOS App Programming Guide - App Design Basics. <https://developer.apple.com/library/ios/documentation/iphone/conceptual/iphonesprogrammingguide/AppDesignBasics/AppDesignBasics.html>, 2013. [Online, accedit el 20-Setembre-2013].
- [6] Cacao. Pàgina oficial de Cacao. <https://cacao.com/getstarted/>, 2013. [Online, accedit el 26-Juny-2013].
- [7] Cedric Vandendriessche. Tutorial on SMS style bubbles. <https://vimeo.com/8718829>, 2013. [Online, accedit el 29-Abril-2013].
- [8] UPC Servei de Comunicació i Promoció. Llibre d'estil dels webs UPC. <http://genweb.upc.edu/feu-servir-genweb/documentacio/guia-destils>, 2013. [Online, accedit el 26-Juny-2013].
- [9] Facebook. Pàgina oficial de Facebook. <https://www.facebook.com/>, 2013. [Online, accedit el 26-Juny-2013].
- [10] Gabinet Jurídic - UPC. Normativa UPC. <http://www.upc.edu/normatives>, 2013. [Online, accedit el 15-Novembre-2013].
- [11] Google Drive. Pàgina oficial de Google Drive. <http://learn.googleapps.com/drive>, 2013. [Online, accedit el 26-Juny-2013].

- [12] Google Plus. Pàgina oficial de Google +. <http://www.google.com/+/learnmore/>, 2013. [Online, accedit el 05-Juliol-2013].
- [13] Hangouts. Pàgina oficial de Hangouts de Google. <http://www.google.com/intl/es/hangouts/>, 2013. [Online, accedit el 26-Juny-2013].
- [14] inLab FIB. inLab FIB talent & tech. <http://inlab.fib.upc.edu/>, 2013. [Online, accedit el 19-Juny-2013].
- [15] Juan Palacio. Flexibilidad con Scrum. Versió online <http://www.navegapolis.net/content/view/694/61/>, Novembre 2007. Creative Commons Reconocimiento-NoComercial 3.0 - [accedit online el 17-Novembre-2013].
- [16] Kirill Fakhroutdinov - uml-diagrams.org. UML Deployment Diagrams Examples. <http://www.uml-diagrams.org/deployment-diagrams-examples.html>, 2013. [Online, accedit el 16-Novembre-2013].
- [17] OAuth community. Estàndard OAuth 2. <http://oauth.net/2/>, 2013. [Online, accedit el 20-Setembre-2013].
- [18] PagePersonnel. Estudio de remuneración tecnología 2012. [http://enginyeriainformatica.cat/wp-content/uploads/2012/10/er\\_tecnologia.pdf](http://enginyeriainformatica.cat/wp-content/uploads/2012/10/er_tecnologia.pdf), 2012. [Online, accedit el 21-Juny-2013].
- [19] RestKit. GitHub de la llibreria RestKit. <https://github.com/RestKit/RestKit>, 2013. [Online, accedit el 20-Setembre-2013].
- [20] STOMP. STOMP Protocol. <http://stomp.github.io/>, 2013. [Online, accedit el 17-Novembre-2013].
- [21] Toni Sellarès - UdG. Introducció a UML. Part III. <http://ima.udg.edu/~sellares/ETIG-ES/IntUML-4.pdf>, 2013. [Online, accedit el 16-Novembre-2013].
- [22] Twitter. Pàgina oficial de Twitter. <https://twitter.com/>, 2013. [Online, accedit el 26-Juny-2013].
- [23] UPCnet. Premi Davyd Luque a la Innovació en les TIC . <http://www.upcnet.es/premi-davydлуque/>, 2013. [Online, accedit el 01-Juliol-2013].
- [24] UPCnet. UPCnet. <http://www.upcnet.es/>, 2013. [Online, accedit el 19-Juny-2013].
- [25] WhatsApp. Pàgina oficial de WhatsApp. <http://www.whatsapp.com/>, 2013. [Online, accedit el 26-Juny-2013].
- [26] Zyncro. Pàgina oficial de Zyncro. <https://www.zyncro.com/ca/>, 2013. [Online, accedit el 08-Juliol-2013].

## 10 Annex

### 10.1 Glossari

**Activitat:** al projecte SomUPC, és un conjunt d'informació que vol publicar un usuari a un context (o un grups de contexts) concrets. Pot ser un text, una imatge o un fitxer qualsevol.

**Activity Stream** [1]: és una especificació per protocols de flux d'activitat, que s'utilitzen per a fer re-difusió de les activitats generades en serveis socials.

**APNS - Apple Push Notification Service:** és un servei creat per *Apple Inc.*, que va ser llançat amb el iOS 3.0 el 17 de juny de 2009. El servei utilitza la tecnologia *push* per enviar notifikacions dels servidors d'aplicacions de tercers als dispositius *iOS*, les notifikacions poden incloure logotips, cançons o alarmes de text personalitzades.

**Context:** al projecte SomUPC, és un conjunt de persones que comparteixen alguna característica en comú, per exemple tots els alumnes d'una assignatura o tots els membres d'una facultat.

**Daily scrum:** reunió diària de 10 minuts de l'equip de desenvolupament, per revisar les tasques realitzades el dia anterior i planificar les tasques a fer durant el dia.

**MVC - Model vista controlador:** patró de disseny per al desenvolupament de programari que separa el model de dades, la interfície usuari i la lògica de control.

**Negoci:** client del producte. Negoci està representat pel *product owner* dintre de *Scrum*.

**Notificació push:** descriu un estil de comunicacions a Internet on la petició d'una transacció s'origina en el servidor. Al contrari a la tecnologia pull, on la petició és originada en el client-servidor.

**Product owner:** representa la veu del client. S'assegura que el resultat del treball de l'equip de Scrum s'adequa des de la perspectiva del negoci. El Product Owner escriu històries d'usuari i les prioritza.

**Scrum:** marc de treball per a la gestió de projectes que defineix un conjunt de pràctiques, on cada persona participant assumeix un rol (*Scrum master*, *Product owner* i equip de desenvolupament), fet que permet adaptar-se a les necessitats i preferències de cada equip o organització.

**Scrum master:** facilitador del *Scrum*, la feina principal del qual és eliminar els obstacles

que impedeixen que l'equip arribi a l'objectiu de cada *Sprint*.

**Sprint:** període en el que es realitza l'increment del producte, idealment aquest període té una durada fixa. La durada fixa té com a objectiu mantenir un ritme constant i facilitar que l'abast dels requeriments associats es respecti per part del client durant la seva execució.

***Sprint planning*:** reunió al inici de cada *Sprint* on es decideix, valora i prioritza les tasques que entren al *Sprint*.

**Subscripció:** acció que realitza l'usuari per rebre al seu *timeline* totes les activitats d'un context.

***Timeline*:** espai on l'usuari pot veure totes les activitats que s'han publicat al sistema als contexts on està subscrit.

**Vista modal:** pantalla que es mostra per sobre de totes les pantalles actives. Si la pantalla que mostra la vista modal té una barra de navegació, la nova vista es superposa i cobreix la pantalla anterior, inclosa la barra de navegació.

## 10.2 Històries d'usuari per *sprint*

A continuació es poden veure les històries d'usuari que s'han realitzat a cada *sprint*. Aquesta tria l'ha fet negoci en funció de les històries que s'han considerat que aporten més valor al projecte.

### ***Sprint 1***

1. Iniciar sessió.

### ***Sprint 2***

26. Tancar sessió.
2. Veure timeline.
3. Veure activitat.

### ***Sprint 3***

14. Veure converses usuari.

### ***Sprint 4***

7. Veure subscripcions.
11. Veure perfil usuari.

### ***Sprint 5***

4. Veure comentaris d'una activitat.
15. Veure missatges d'una conversa.

### ***Sprint 6***

5. Crear una activitat.

### ***Sprint 7***

6. Afegir un comentari.
13. Modificar dades perfil usuari.
16. Afegir un nou missatge a una conversa.

### ***Sprint 8***

17. Visualitzar dades d'una conversa.
18. Modificar dades d'una conversa.

### ***Sprint 9***

19. Crear una conversa.

### ***Sprint 10***

8. Veure activitats d'una subscripció.

12. Veure activitats d'un usuari.

### ***Sprint 11***

9. Afegir una subscripció.

10. Cercar contextos públics.

### ***Sprint 12***

22. Esborrar una activitat.

23. Esborrar un comentari.

24. Esborrar una subscripció.

25. Sortir/Esborrar una conversa.

### ***Sprint 13***

27. Publicar un arxiu a una activitat.

### ***Sprint 14***

20. Notificacions de converses.

### ***Sprint 15***

21. Converses en temps real.

### ***Sprint 16***

Arreglar problemes i acabar coses pendents.

## 10.3 Llista d'ObjectTypes

Llistat 1: ObjectType activity

```
1 {
2   "generator": null,
3   "contexts": [{
4     "url": "http://atenea.upc.edu/local/max/max/goto.php?path=/1/167/4
5       24",
6     "hash": "7615301bea3d2c6f35289cae3eeba4fa073d634d",
7     "displayName": "Introducci\u00f3n a la atenci\u00f3n efectiva al
8       cliente",
9     "objectType": "context"
10  }],
11  "object": {
12    "content": "Dia 10blended cajero",
13    "objectType": "note"
14  },
15  "actor": {
16    "username": "william.learn",
17    "displayName": "William Learn",
18    "objectType": "person"
19  },
20  "commented": "2013-11-07T22:31:05Z",
21  "published": "2013-11-07T22:31:05Z",
22  "verb": "post",
23  "deletable": true,
24  "replies": [],
25  "id": "527c14a94496402db22fd7c6",
26  "objectType": "activity"
27 }
```

Llistat 2: ObjectType person

```
1 {
2   "username": "william.learn",
3   "iosDevices": ["646a298e8b048bf105....d3f05c6ef976391d0"],
4   "displayName": "William Learn",
5   "talkingIn": [],
6   "creator": "restricted",
7   "id": "525f994f4496401b809afa31",
```

```

8   "androidDevices": [],
9   "owner": "restricted",
10  "subscribedTo": [],
11  "last_login": "2013-10-17T08:01:19Z",
12  "published": "2013-10-17T08:01:19Z",
13  "following": [],
14  "objectType": "person"
15 }

```

### Llistat 3: ObjectType context

```

1 {
2   "displayName": "Habilidades directivas I: gesti\u00f3n de la
      resistencia y la planificaci\u00f3n",
3   "tags": [],
4   "url": "http://atenea.upc.edu/max/max/goto.php?path=/1/146/437",
5   "creator": "restricted",
6   "id": "527b77c54496402db22fd78e",
7   "published": "2013-11-07T11:21:41Z",
8   "owner": "restricted",
9   "hash": "578345a23a56daf7a09a0a0b1ff45dccc48b4206",
10  "permissions": {
11    "read": "public",
12    "write": "subscribed",
13    "invite": "restricted",
14    "subscribe": "public"
15  },
16  "objectType": "context"
17 }

```

### Llistat 4: ObjectType conversation

```

1 {
2   "displayName": "steve.talk, william.learn",
3   "creator": "william.learn",
4   "messages": 1,
5   "id": "526e63e0449640094a598fcf",
6   "objectType": "conversation",
7   "participants": ["steve.talk",
8     "william.learn"],
9   "lastMessage": {

```



```

10     "content": "Test",
11     "published": "2013-10-28T13:17:20Z"
12 },
13 "published": "2013-10-28T13:17:20Z",
14 "owner": "william.learn",
15 "permissions": {
16     "read": "subscribed",
17     "write": "subscribed",
18     "unsubscribe": "public",
19     "invite": "restricted",
20     "subscribe": "restricted"
21 }
22 }

```

#### Llistat 5: ObjectType message

```

1 {
2     "generator": null,
3     "contexts": [{
4         "participants": ["steve.talk",
5             "william.learn"],
6         "displayName": "steve.talk,
7             william.learn",
8         "id": "526e63e0449640094a598fcf",
9         "objectType": "conversation"
10    }],
11     "object": {
12         "content": "Test",
13         "objectType": "note"
14    },
15     "actor": {
16         "username": "william.learn",
17         "displayName": "William Learn",
18         "objectType": "person"
19    },
20     "commented": "2013-10-28T13:17:20Z",
21     "published": "2013-10-28T13:17:20Z",
22     "verb": "post",
23     "replies": [],
24     "id": "526e63e0449640094a598fd2",

```

```
25     "objectType": "message"
26 }
```

#### Llistat 6: ObjectType comment

```
1 {
2     "content": "En Peru",
3     "deletable": true,
4     "published": "2013-11-08T16:15:32Z",
5     "id": "527d0e244496402db22fd7d6",
6     "actor": {
7         "username": "william.learn",
8         "displayName": "William Learn",
9         "objectType": "person"
10    },
11     "objectType": "comment"
12 }
```

## 10.4 MAX

MAX és un sistema de recollida i visualització de registres d'activitat generada per usuaris i aplicacions així com les interaccions entre usuaris i entre usuaris i aplicacions.

El MAX registra l'activitat de dues maneres: Activa i passiva. Quan ho fa de manera passiva, els usuaris i aplicacions que generen activitat, usen l'API del MAX per registrar-hi activitat. Quan ho fa de manera activa, el MAX va a buscar l'activitat al sistema que la genera a través d'unes regles predefinides, per exemple, l'activitat generada per un compte de Twitter i un hashtag associat.

Tant els usuaris com les aplicacions que puguin interactuar amb el MAX poden ser d'origen en escenaris corporatius (interns) o en canvi ser totalment externs al sistema.

Aquest sistema disposa d'una API REST que està protegida amb OAuth2. Aquesta API està documentada a <http://max.beta.upcnet.es/docs/v3/ca/>.

El codi del projecte MAX està disponible al GitHub <https://github.com/UPCnet/maxserver>